Zimmerman, Kiana M., A Spatial Model for Markov Chain Analysis of Grasshopper Population Dynamics in Wyoming, M.S., Department of Computer Science, December, 1999.

When landowners/managers make decisions about how to try to control grasshopper infestation, they need to take into consideration grasshopper population dynamics. The goal of this project was to design a prototype extension to run in ArcView that applied a two-state Markov Chain to grasshopper infestation maps in Wyoming from 1944 to 1996 to predict future population dynamics. The prototyped extension within an existing Geographic Information System was able to predict one-year infestation probabilities, length of the infestation, and multiyear infestation probabilities. Maps were produced that show the likelihood of changes in infestation status and length of infestation with relatively fine-grained analysis, 1,000 m^2, for Wyoming. The series of maps can be used to predict the probabilities of being infested one-year given the previous year's conditions and how long an infestation may last. This information is helpful in making management decisions, such as deciding whether insecticide will or will not be used to treat the grasshopper infestation.

A SPATIAL MODEL FOR MARKOV CHAIN ANALYSIS OF GRASSHOPPER POPULATION DYNAMICS IN WYOMING

By
Kiana Marie Zimmerman

A thesis submitted to the Department of Computer Science
and the Graduate School of the University of Wyoming
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER SCIENCE

Laramie, Wyoming
December, 1999

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Scott Schell who suggested that I talk to Dr. Jeffrey Lockwood about a project and Dr. Lockwood for having such an interesting project.

I would like to thank my committee members, Drs. Gantenbein, Bauer, and Lockwood.

I would also like to thank my husband, Martin, my mom, Ila, and my brother, Jared, for being so great.

Chapter 1

Introduction

Humans have been at war with grasshoppers and locusts since the beginning of recorded history [1]. Grasshoppers and locusts compete with humans for crops and with livestock for rangeland forage [2]. In the western United States, there are approximately 400 species of grasshoppers [2], while in Wyoming there are over 100 species [3]. Of the 400 species 350 are not an economic concern; they are innocuous or beneficial [4]. However, when the remaining species reach high densities they can severely deplete rangeland forage [2]. Hewitt et al. [5] estimate that 25% of the available rangeland forage is eaten by grasshoppers. In 1997, the United States Department of Agriculture Animal and Plant Health Inspection Service Plant Protection Quarantine (USDA-APHIS-PPQ) eliminated its cost-share program for treating grasshopper infestations. This along with inflation effectively tripled the cost of treatment to landowner/managers [6]. In 1985-1986, $22.75 million was spent to try to control grasshoppers in Wyoming [2].

With shrinking resources to treat grasshoppers, it is important for the landowner/manager to make informed decisions on how to deal with grasshopper outbreaks. Some entomologists [7] believe that grasshopper infestations can be treated with insecticide; based on the rate that the annual grasshopper population increase, the landowner/manager may experience multiyear benefit beyond the year of treatment. However others disagree [8]. The answer to the question, "how long will an infestation persist" is the single most important parameter in economic models of grasshopper control programs. In any event, the landowner/manager making decisions about how to treat a grasshopper outbreak should ask;

- Given that there is an infestation this year, what are the chances of an infestation next year?
- If I treat, will I receive multiyear benefit?
- How long will the infestation last?

It is important to analyze where and how to best allocate the existing resources for grasshopper management. One important factor in grasshopper management is predicting the areas of future outbreaks. Computers and mathematical techniques are valuable tools in such predictions. The goal of this work was to produce a prototype computer program to predict future grasshopper infestation probabilities and lengths of infestations.

The management product derived from this analysis was a set of maps showing the likelihood of changes in infestation status (uninfested to uninfested, uninfested to infested, infested to infested, and infested to uninfested) of Wyoming lands. To assure that the maps were useful to individual landowners/managers a grid cell size of 1,000 meters was used to show the transitions, duration, and multiyear dynamics.

A Geographic Information System (GIS) was used by Schell and Lockwood [9] to map the historical frequency of grasshopper infestations in Wyoming. A GIS handles geographically referenced data as a visual form (maps). The prototype made use of Schell's geographically referenced data to try to predict the course of infestations using a two-state Markov chain. Markov chains are a means of analyzing stochastic processes allowing prediction of the future of a system based on its current state [10].

Chapter 2 gives a description of how Geographic Information Systems have been used in agriculture and rangeland management to help make management decisions.

Chapter 3 provides an introduction to stochastic processes and describes the two-state Markov model that was chosen to model the grasshopper infestations in Wyoming. This chapter also reviews the application of two-state Markov chains to predicting grasshopper infestations.

Chapter 4 describes the prototype that was developed and tested to implement the two-state Markov model. Chapter 5 evaluates the prototype by comparing hand-simulated values with the prototype's predictions and chapter 6 concludes the work.

Chapter 2

Geographic Information Systems

The goal of the project was to use basic GIS tools (allowing the user to zoom in and out, displaying maps and other manipulations) combined with a two-state Markov chain to produce a set of maps that allow the user to show the transition probabilities, duration, and multiyear dynamics of grasshopper infestation. The prototype system used historical grasshopper infestation data for Wyoming gathered by Schell et al. [9] in GIS format. Rather than creating a GIS from scratch it was decided to add functionality by enhancing an existing GIS, ArcView*. This chapter will describe Geographic Information Systems and how this technology has been used in agriculture and rangeland management.

A GIS is an organized collection of computer hardware and software, data, personnel, organizations and institutions with the goal of collecting, storing, updating, manipulating, analyzing, sharing, and displaying geographically referenced information [11, 12].

GISs can be divided into two categories, raster and vector. A raster GIS uses cells to represent entities in the real world. Each cell is assigned a numerical value that represents information about the real world. For instance, a cell with an 89 might mean that a lake is in that location. All of the cells corresponding to the location of a lake have 89 for their value. A vector GIS represents the world with points, lines, and polygons.

---

* ArcView is a trademark of Environmental Systems Research Institute, Inc.

The lake would be represented in the vector GIS with a polygon conforming to the outline of the lake.

GISs have been used in many unique ways to formulate answers to questions on how to best manage agricultural resources. With a GIS, questions about spatial information can be answered, scenarios can be run to help with management decisions, and events can be modeled. In China, Zesheng and Ling [13] created a GIS that allowed farmers to manage their weed infestations. The management process began by mapping the weed distribution in the field. Once the weeds were mapped, the herbicide and dose were chosen from a database. Finally, a map was produced to help with the herbicide application. This allowed the farmer only to treat areas where weeds were present, reducing the amount of chemicals placed in the environment and reducing the costs to produce the crop.

In Great Britain, farmers are paid to take part of their land out of production for 1 to 5 years to help improve the biodiversity of the area. Swetnam et al. [14] created a GIS that helps predict the weeds that will grow in the land that is set aside. The system also helps predict the effects of different control methods.

The Deaver Vineyard used a desktop mapping program [15] to manage their entire production of wine. They integrated a GIS into the mapping of all of their vine locations, including information about each vine (date planted, rootstock, variety, outbreaks of diseases or pathogens, and insect trap locations), topography of the land, soil type, growing season, degree-days, irrigation system locations, wind machine locations, machinery locations and maintenance records. They were able to run different scenarios based on the databases in the GIS to select the best grapes for the different locations on their property. They produced yield maps that could be used in the next year's decisions. Maps were also produced for employees and contractors who needed to know where something of particular interest was located. Based on the maps, they could even validate the taxes assessed on their land. This system has helped them reduce costs, produce better wines, and reduce paperwork.

With the help of Global Positioning Systems (GPSs), GISs, remote sensing, on-the-go sensors, monitors, and controllers, farmers are now better able to know what is taking place in all parts of a field. This management strategy has been termed "precision agriculture." The field is no longer thought of as one homogenous entity, but instead as a field with many needs in different locations. This allows the farmer to cut costs in many ways, such as using herbicides and fertilizer in a more appropriate manner [16].

GIS has many applications beyond farming, however. In rangeland, GIS has been used to suggest the number of cows (stocking rate) that should be allowed to graze in environmentally critical areas, such as watersheds. Databases showing the topography, soils, residual dry matter requirements, production, rainfall, and sensitive habitats have been integrated using a GIS [17]. The databases can then used to suggest the number of cows that can graze in a field.

A GIS has been used to predict the movement of grizzly bear, elk, and cougar in Montana [18]. Based on the habitat, maps were developed to locate the possible paths these animals could take in moving from place to place. They were able to identify locations where special management decisions needed to be made for the welfare of the animals, provided that the animals used the computer-forecasted path.

In Saguaro National Park, a GIS [19] has been used to map past fires and their attributes and analyze fire patterns. Areas of high risk were located for fire management in the future.

oviposition, breeding, hatching, and developing, 2) Investigate how grasshoppers and the environment interact, and 3) Map populations and areas that are predisposed to infestations. But GIS technology has not been applied to predicting population dynamics or future infestations. Several questions need to be answered in grasshopper management. For instance, if land is not infested what is the likelihood it will stay uninfested or become infested? If it becomes infested, how long will it stay infested? Based on these population dynamics, should the land be treated with insecticides to try to control the grasshoppers? Using geographically referenced data in the form of electronic maps, a GIS can be used to model and forecast grasshopper population dynamics. The past populations are converted to spatial data that can be analyzed, summarized, and displayed in a GIS.

What is needed, however, is a way to predict future populations based on the past. The next chapter describes a mathematical model that can be used for such a prediction.

Chapter 3

Stochastic Processes and Two-State Markov Chains

In this project, information on future infestations will be drawn from grasshopper infestation maps from 1944 to 1996 in Wyoming. Statistics can be used to gather more information from the data and to predict the future based on the past events. A stochastic, or chance, process, is a mathematical model that is probabilistic and is best suited to a collection or a family of random variables indexed by time and space [25]. This chapter defines stochastic processes in general and a particular stochastic process called a two-state Markov chain.

Stander et al. [26] refer to a process as a series of events and a stochastic process as one that is not determined by its initial state. Stewart [10] defines a stochastic process as a family of random variables $\{X(t), t \in T\}$ defined on a given probability space and indexed by a parameter, t. The value of t varies over some index set, usually time or space [25]. A stochastic process is a good model for a time series, according to Bhat [25]. A process that has a discrete parameter and state space is called a Markov chain [25]. Based on a given series of events or states, a Markov chain predicts the probability of being in a state at a specific time and the time taken until a state is first reached [10].

Markov chain analysis has been used in several applications. It has been applied to criminology, job performance, baseball strategy, livestock disease management, hydrogeological properties, and flour milling forecasts. Stander et al. [26] tried to predict the types of offenses committed by a criminal based on the convict's previous convictions. It was discovered that the offences committed are influenced by present and previous offenses. Criminals tend to specialize, and therefore a first-order (one-step Markov transition) Markov chain does not adequately predict offenses. However, a second-order (two-step Markov transition) Markov chain may be able to predict the offenses. Stander et al. also applied Markov chains to sentencing, and it was shown that sentencing is influenced by past sentencing.

It was shown by Montgomery et al. [27] that the effects of administration of faculty members could be analyzed using Markov chain analysis. Using a spreadsheet and the Markov chain model they were able to run several scenarios that could examine how hiring, promotion, retirement, tenure ratios, and appointment distribution would affect the college.

Pankin [28] used Markov chain analysis and a spreadsheet to estimate the effects of batting order on the batting average in the game of baseball.

Markov chains have also been used in the prediction of diseases in livestock. Markov chains were used within a larger model written in Turbo Pascal to analyze economic choices when cattle and pigs have foot-and-mouth disease [29]. A model was designed that simulated the spread of the disease and the costs of different management practices. The computer program that was developed was a user-friendly model that allowed the manager to enter different epidemiological and economic information to determine the results of treating or not treating foot-and-mouth disease.

Markov chains [30] were integrated with epidemiologic modeling of cattle diseases. A spreadsheet was used to estimate the number of cattle not infected, infected with *Streptococcus agalactiae*, infected with other *Streptococcus* spp., infected with *Staphylococcus*, infected with another mastitis causing agent, or culled from the herd.

Carpenter showed using his spreadsheets that the herd stabilized by the fourth or fifth year when the herd contained cows in all of the above states a result consistent with dynamics of an endemic disease.

Rosen et al. [31] used Visual Basic and a GIS called IDRISI to create a model using Markov chain information about hydrogeological and geological properties to determine the best locations for a nuclear waste repository. A Bayesian Markov Geostatistical Model was created that could be used to find a location in a region that might be suitable for the site, within a site an exact location for the repository, and/or detailed investigations in the construction phase. The model was designed to use the data currently available but could also allow data to be integrated, as it became available.

Markov chain analysis was applied to the flour milling industry to explore what causes the number and size of flourmills to change and to determine if there will be enough wheat flour in the year 2000 [32]. Somewhat paradoxically, the analysis showed that the number of mills decreased but the flour production increased. After running three different scenarios through the Markov model, the researcher decided that rising disposable income and declining wheat prices were the reasons the number of mills will decrease, but that there will probably be enough flour in the year 2000.

A Markov chain was applied to grasshopper population dynamics in Saskatchewan, Montana, and Wyoming [33-35]. Infestation maps were created using yearly roadside grasshopper survey data. If any location in the selected area had more than ≥ 9.6 grasshoppers per m^2 the entire area of interest was considered infested. Markov chain analysis was used in the United States to answer the following questions:

- If the county has low/high infestation this year, what is the probability it will be low/high next year?
- What is the probability in future years of infestation remaining low/high?
- What is the expected number of years the infestation will remain low/high?
- What is the expected number of years before the state will switch to the other infestation state?

The way this data was displayed varied. Mukerji et al. [33] used three-dimensional plots to visually display and analyze their data. They represented the georeferenced data by using the latitude and longitude as the X and Y axes and population density as the Z-axis with three-dimensional software. The X and Y-axis were shifted to shape a rectangle similar to the geographical shape of Saskatchewan. In contrast, Kemp and Lockwood's [34, 35] method involved hand simulation.

Analyzing Grasshopper Population Dynamics with a Two-State Markov Chain

These approaches allowed estimations of probability, stability, and duration of grasshopper outbreaks [35], but only on a coarse scale. The models worked on the county level. An entire county was considered infested if any location had ≥ 9.6 grasshoppers per m^2; this model overestimates the infestation for the county. With a more accurate spatial tool, the landowners/managers could make a more informed decision about treating a grasshopper outbreak. Environment plays an important role in grasshopper outbreaks and this complex of factors is not taken into consideration when the entire county (a political, not ecological unit) is used as the spatial unit.

A series of events can be represented in a GIS in the form of an electronic raster-based map for each year because grasshoppers are univoltine (one generation per year).

A value can be assigned to each raster cell of the map to represent the state at a particular time. The transition probabilities for the system can be analyzed along with other ecological factors to make management decisions. Using a GIS and Markov chain analysis, a more accurate representation of future grasshopper infestations can be developed.

Every Markov chain has a set of mutually exclusive states [10]. The transition from state to state is considered instantaneous, meaning it takes zero time. Furthermore, the future of the system only depends on the current state and not on the past states [10]. When there are only two states in a Markov chain, it is called a two-state Markov chain. Two-state Markov chains are intuitive, easy to evaluate, and appropriate when there are only two outcomes or states [25].

Using a Markov chain we can answer the following questions: 1) What is the expected duration of each state? 2) What are the mean and variance of each state? and 3) What state is preferred in the long run?.

We are interested in the behavior of grasshopper infestation at locations throughout Wyoming. Each location in Wyoming can be in one of two different states, uninfested or infested. A location is considered infested if there are ≥ 9.6 grasshoppers per m^2 ($8/yd^2$) [9]. Some entomologists consider ≥ 9.6 grasshoppers per m^2 the economic injury level, which means that damage caused by the grasshopper becomes equal to the cost of controlling it, but this does not take into consideration rangeland productivity, forage and livestock prices, and grasshopper species composition [36]. However, this population density has ecological meaning as the rangeland carrying capacity [6].

I chose a two-state Markov chain over other stochastic processes because: 1) It is conceptually simple, 2) It had been previously used in the context of analyzing grasshopper dynamics, 3) The results of the analysis are relevant to management (transitional probabilities, expected duration, etc), 4) It is easy to derive from successional data, and 5) The results of the analysis are readily adaptable to graphical representation [37, 38].

The Wyoming Department of Agriculture (WDA) and the USDA APHIS PPQ collaborated in making grasshopper density observations every year from 1941 to 1980. The observations were discontinued from 1981 to 1984, then reinstated for 1985 to 1996 by. Therefore, we have a discrete parameter space set T = {0,1,2,3…} where time 0 = 1941, 1 = 1942, etc. As such this data set is a discrete- (time) parameter stochastic process or Markov chain.

At any time t, the cells of the maps are in either 0 (uninfested) or 1 (infested) states. Therefore, the state space is {0,1}. Since the system to be modeled has two states, 0 and 1, a two-state Markov chain was used. There are four possible transitions: uninfested →uninfested (0 →0), uninfested →infested (0 →1), infested →uninfested (1 →0), and infested →infested (1 →1).

The probabilities of each transition were determined according to [39]:

$$P(\text{Uninfested} \longrightarrow \text{Infested}) = \frac{\text{Number of Transitions from 0 to 1}}{\text{Total Number of Transitions Starting with 0}}$$

$$P(\text{Infested} \longrightarrow \text{Uninfested}) = \frac{\text{Number of Transitions from 1 to 0}}{\text{Total Number of Transitions Starting with 1}}$$

It was only necessary to compute the transition probabilities for P_{01} and P_{10} and use those values to derive the other two probabilities.

$$P(\text{Uninfested} \longrightarrow \text{Uninfested}) = 1 - P(\text{Uninfested} \longrightarrow \text{Infested})$$

$$P(\text{Infested} \longrightarrow \text{Infested}) = 1 - P(\text{Infested} \longrightarrow \text{Uninfested})$$

If the probability is 0, the transition is called impossible; if the probability is 1, the transition is called certain or recurrent; and if the probability is $0 < X < 1$, the state is transient [10, 40].

Let $\{X_n, n = 0,1,2,...\}$ be a Markov chain and S be the state space with the states

$$\{0,1\}$$

Therefore, the transition probabilities (n-step probabilities) are [34]:

$$P_{ij}^{(m,n)} = P(X_n = j \mid X_m = i) \quad i, j = 0,1; m \leq n$$

and [33]

$$X_n = \begin{cases} 0 & \text{Low Density} \\ 1 & \text{High Density} \end{cases}$$

for n = 1,2,…

and the transition matrix is [39]

$$P = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix} \quad 0 \leq a, b \leq 1$$

Each probability must be:

$$0 \leq \text{probability} \leq 1 \text{ [25]}$$

and the summation of the rows must equal 1:

$$\sum_{\text{all } j} P_{ij} = 1 \text{ for all i [10].}$$

To determine the transition probabilities for n-steps greater than 1 [25, 38]:

$$P^{n} = P^{n-1}P$$

or [34]

$$P_{00}^{(n)} = P_{00}^{(n-1)}P_{00} + P_{01}^{(n-1)}P_{10}$$

$$P_{01}^{(n)} = P_{00}^{(n-1)}P_{01} + P_{01}^{(n-1)}P_{11}$$

$$P_{10}^{(n)} = P_{10}^{(n-1)}P_{00} + P_{11}^{(n-1)}P_{10}$$

$$P_{11}^{(n)} = P_{10}^{(n-1)}P_{01} + P_{11}^{(n-1)}P_{11}.$$

The expected number of years a location will stay in one state before switching to another state or occupation time is calculated by [34]:

$$E(\alpha_0) = \frac{1-a}{a} \quad \text{and} \quad E(\alpha_1) = \frac{1-b}{b}$$

with variances

$$V(\alpha_0) = \frac{1-a}{a^2} \quad \text{and} \quad V(\alpha_1) = \frac{1-b}{b^2}.$$

and confidence interval [6]:

$$\delta = \sqrt{V(\alpha)}$$

$$95\%CI = \frac{1.96 * \delta}{\sqrt{n}}.$$

To determine the number of times a cell is in each state:

$$\mu_{00}^{(n)} = \sum_{k=1}^{n} \frac{b}{a+b} + \frac{a(1-a-b)^{k}}{a+b}$$

$$\mu_{01}^{(n)} = \sum_{k=1}^{n} \frac{a}{a+b} - \frac{a(1-a-b)^{k}}{a+b}$$

$$\mu_{10}^{(n)} = \sum_{k=1}^{n} \frac{b}{a+b} - \frac{b(1-a-b)^k}{a+b}$$

$$\mu_{11}^{(n)} = \sum_{k=1}^{n} \frac{a}{a+b} + \frac{b(1-a-b)^k}{a+b}$$

Finally, the duration that a cell is in each state is determined by [39]

$$\left(\frac{1}{n}\right)\mu_{ij}^{(n)}$$

This statistical model was used in a prototype tool to predict grasshopper population dynamics based on GIS data, as described in the next chapter.

Chapter 4

Prototyping an ArcView Extension

The purpose of this work was to determine if a two-state Markov chain implemented with a GIS could be used to predict grasshopper population dynamics. This approach was used to integrate a two-state Markov chain within a GIS and to test whether the method accurately predicts grasshopper population dynamics.

Prototyping

Prototyping is a method to investigate if a system can be implemented and used to solve a presented problem [41]. It is a quick way to design all or part of a system. The iterative nature of prototyping allows all aspects of a system to be explored. The designer of the system can get a better feel for the user's requirements and whether the problem can be solved. It also allows the users to review the program's functionality to decide if it fulfills their desired results.

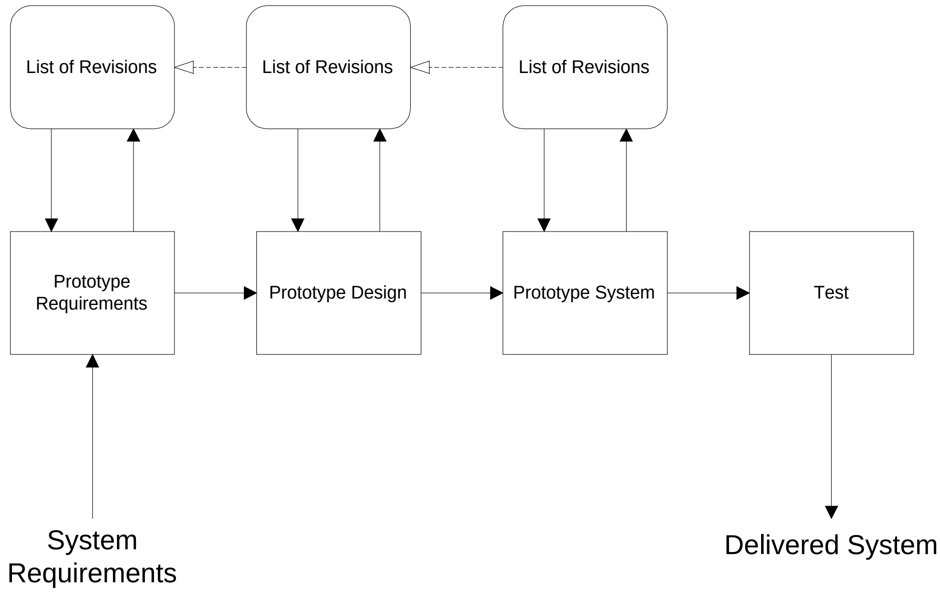


Figure 1. The Prototype Model As Defined By Pfleeger

Figure 1 shows the prototyping steps according to Pfleeger [41]. The steps of prototyping are: 1) Determine the system's requirements, 2) Build the prototype requirements based on the initial requirements, 3) Design the prototype, 4) Prototype the system, 5) Test, and 6) Deliver the system. Steps two through four can be repeated many times to create a system that is satisfactory to the end user, as shown in Figure 1.

Prototyping the Two-State Markov Chain

To implement the two-state Markov chain model of the grasshopper infestations, a prototype was created in ArcView. Using the Spatial Analyst Extension and Avenue*, an object-oriented programming language that allows the user to add functionality and customize ArcView, a two-state Markov chain was prototyped and tested within a vector GIS using an extension to add raster capabilities.

* Spatial Analyst and Avenue are trademarks of Environmental Systems Research Institute, Inc.

The entire system was created using the prototyping model depicted in Figure 1. The first step was to define the system's requirements, which were to take geographically referenced data and perform the two-state Markov chain analysis. In step two, it was decided that the entire system would be implemented to see if the GIS could perform all aspects of the two-state Markov chain to predict grasshopper population dynamics.

The third step required designing a system to work within an existing GIS. ArcView was chosen over other GISs because:

- the University of Wyoming currently has a site license,
- the information gathered by Schell [9] could be imported,
- this GIS gave the framework that allowed functionality to be added, and
- ArcView gave the user the basic GIS functionality like zooming.

The system was prototyped as an extension to ArcView. An extension in ArcView can be loaded and unloaded at the user's discretion. All of the added functionality is housed within the extension. The fourth step of testing was performed on data gathered by Schell [9]. At all stages of the prototyping method, Dr. Jeff3(c)(hell)91iS fd grasshopper population density estimation were taken [for a description of the survey see 42]. Surveys for grasshoppers are usually done on the eastern part of Wyoming. Hand drawn maps were then created based on a “subjective interpolation of the survey point data.” The survey method of WDA used is unknown, but is assumed to be the similar to that of USDA-APHIS-PPQ [9]. USDA-APHIS-PPQ supplied the data for the maps after 1985, and WDA supplied the data for the maps via USDA-APHIS-PPQ before 1985 [9]. Maps from 1944 to 1980 and 1985 to 1996 were gathered (data from 1981-1984 is missing because no survey was performed).

The information from the hand-drawn maps was digitized into a raster GIS, ERDAS*. A grid cell size of 1,000 m^2 was used. Each cell was coded with a 0 if the cell was uninfested or 1 if the cell was infested with ≥ 9.6 grasshoppers per m^2 (this was the only density consistently recorded throughout the survey period) [9]. In ERDAS, the maps were stored in Lambert Conformal Conic projection [43]. The data were first exported out of ERDAS and imported into ArcView using a raster extension, Spatial Analyst. Each year’s data were represented as a separate grid.

ArcView Extension

This two-state Markov chain model was prototyped in ArcView with the extension Spatial Analyst. ArcView was used because it gives the basic GIS operations and allows scripts to be written in Avenue, allowing the functionality of ArcView to be extended and altered.

---

* ERDAS is a trademark of Erdas, Inc.

The prototyped Markov extension adds a menu option called "Markov Chain" to the basic ArcView menu bar. From this menu choice the user can perform the steps of the Markov chain analysis to answer the questions

- Given that there is an infestation this year, what are the chances of an infestation next year?
- If I treat, will I receive multiyear benefit?
- How long will the infestation last?

Description of the Extension's Functionality

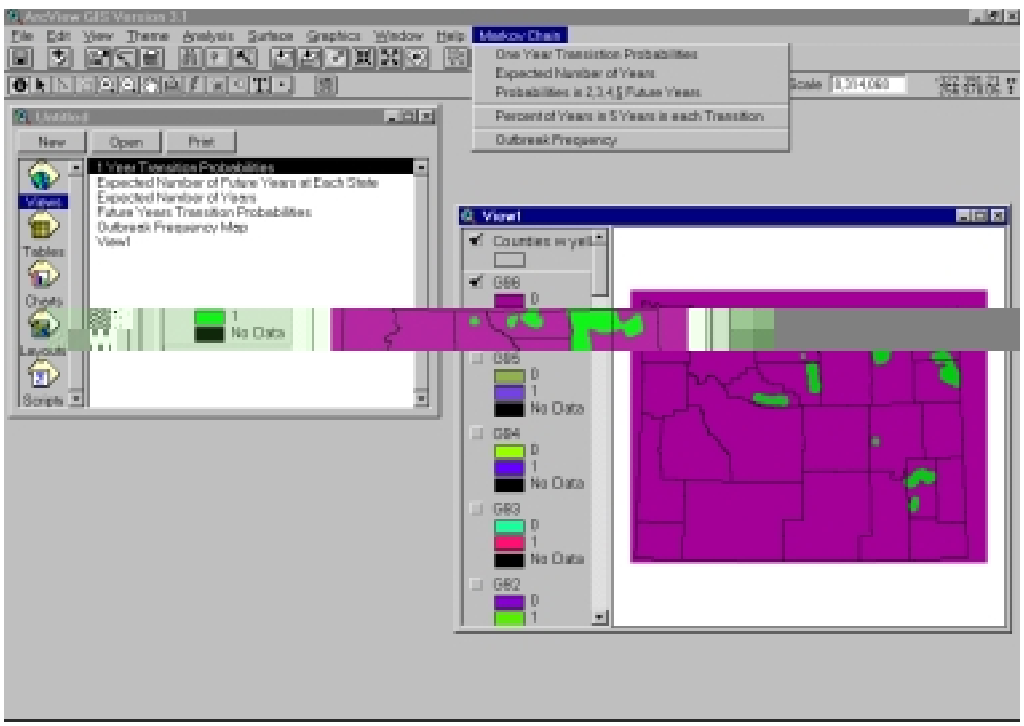


Figure 2. Screenshot Of The Markov Chain Extension Within ArcView

Figure 2 shows a screenshot of ArcView with the two-state Markov chain extension. The Markov chain menu choice is available after the user has loaded the extension. Once the extension is available the user opens a view and adds all the grids that represent the states in a series of years. The states are 0 and 1, and each year is represented as a grid. To analyze infestations from 1944 to 1980 the user needs 37 grids (36 transitions). From the raw data set, the user can perform the "Outbreak Frequency" or "One Year Transition Probabilities" analysis.

Outbreak Frequency

The outbreak frequency analysis is based on Schell's work [9]. It computes the frequency with which each cell was infested with ≥ 9.6 grasshoppers per m^2 by summing up the values in each cell over the time period.

One Year Transition Probabilities

The menu choice “One Year Transition Probabilities” creates four grids, in a new view, that represent the one-step transition matrix, $P_{00}, P_{01}, P_{10},$ and P_{11}. These four grids are used to answer “If a cell is uninfested/infested this year, what are the chances it will be uninfested/infested next year?”

This menu choice sorts the grids into alphabetical order; it was assumed that the user would name the grids similar to g1, g2, and so on. The grid g1 would represent the cell values for year 1. The prototype then creates temporary grids that track the number of transitions from 0 to 1 and 1 to 0. Temporary grids are also used to count the number of transitions that started in 0 and 1. For example, to compute the probability of the transition from 0 to 1 (P_{01}) the temporary grid tracking the number of transitions from 0 to 1 was divided by the temporary grid tracking the number of transitions that started at 0. To compute the transition from 0 to 0, the previously computed grid was subtracted from 1. The resulting grids are displayed in a view and labeled.

Expected Number of Years

Once the four, one-step transition grids have been computed, the user can then use the menu option “Expected Number of Years” to compute the time that each cell will be in each state before switching to the other state. The grids computed by "One Year Transition Probabilities" are used to compute the expected number of years, variance, and 95% confidence interval grids. The grids representing P_{01} and P_{10} ($\mathrm{a} = \mathrm{P}_{01}$ and $\mathrm{b} = \mathrm{P}_{10}$) are used in the following equations. A new view is created that displays the computed grids. These grids can be used to answer the question “How long will the uninfested or infested conditions last?”

Using grid subtraction, division, and power, the following values for expected duration were computed and displayed as grids

$$E(\alpha_0) = \frac{1-a}{a} \quad \text{and} \quad E(\alpha_1) = \frac{1-b}{b}$$

and variances

$$V(\alpha_0) = \frac{1-a}{a^2} \quad \text{and} \quad V(\alpha_1) = \frac{1-b}{b^2} \ [25].$$

and confidence interval [6]:

$$95\%\mathrm{CI} = \frac{1.96 * \delta}{\sqrt{\mathrm{n}}}.$$

Probabilities in 2, 3, 4, and 5 Future Years

To compute the probability of each cell being in state 0 or 1 at times beyond the next year, the user would use the menu option “Probabilities in 2, 3, 4, 5 Future Years.” This option computes the n-step transition matrix in the form of sixteen grids and displays the information in a new view. The sixteen grids that represent the four transition matrices are computed by taking the grids computed by the "One Year Transition Probabilities" and multiplying them in a matter to satisfy

$$P_{00}^{(n)} = P_{00}^{(n-1)}P_{00} + P_{01}^{(n-1)}P_{10}$$

$$P_{01}^{(n)} = P_{00}^{(n-1)}P_{01} + P_{01}^{(n-1)}P_{11}$$

$$P_{10}^{(n)} = P_{10}^{(n-1)}P_{00} + P_{11}^{(n-1)}P_{10}$$

$$P_{11}^{(n)} = P_{10}^{(n-1)}P_{01} + P_{11}^{(n-1)}P_{11} \text{ [34].}$$

Percent of Years in 5 Years in each State/Condition

The last menu option “Percent of Years in 5 Years in each Transition” creates a view that displays the number of years and percent of time each cell is expected to be in each state in the next five years, given that it started in 0 or 1. The grids computed by the menu choice "Probabilities in 2, 3, 4, 5 Future Years" are used to compute:

$$\mu_{00}^{(n)} = \sum_{k=1}^{n} \frac{b}{a+b} + \frac{a(1-a-b)^k}{a+b},$$

$$\mu_{01}^{(n)} = \sum_{k=1}^{n} \frac{a}{a+b} - \frac{a(1-a-b)^k}{a+b},$$

$$\mu_{10}^{(n)} = \sum_{k=1}^{n} \frac{b}{a+b} - \frac{b(1-a-b)^k}{a+b},$$

$$\mu_{11}^{(n)} = \sum_{k=1}^{n} \frac{a}{a+b} + \frac{b(1-a-b)^k}{a+b},$$

$$\text{and } \left(\frac{1}{n}\right)\mu_{ij}^{(n)} \text{ [34, 39].}$$

Given this prototype and GIS data, a user can produce predictions about grasshopper population dynamics in a number of ways. In the next chapter, the prototype's predictive ability is evaluated by comparing its outputs to hand-calculated and real values.

Chapter 5

Evaluation

An extension was prototyped in ArcView to predict grasshopper infestations. Two methods were chosen to test the prototype. The first test compared hand-simulated values with the prototype to see if the prototype produced expected results. The second test compared values produced by the prototype to the real world to see if the prototype could accurately predict real world events. In the first test, seven sites were tabulated and a hand/computer simulation was run. The results were then compared against the results from the prototype extension. To test how the two-state Markov chain compares to real world events, the prototype was run on data from 1944 to 1975, and seven sites were chosen to compare against the actual values in 1976 to 1980. The test cases showed that the extension works. With the results from the created prototype, a landowner/manager can make more informed decisions about grasshopper control in Wyoming.

Site Location / Year	1 -12,300,077,000	2 79,000,171,000	3 -87,000,235,000	4 253,000,373,000	5 255,000,164,000	6 271,000,221,000	7 27,000,420,000
1944	0	0	0	0	0	0	0
1945	0	0	1	0	0	0	1
1946	0	0	0	0	1	0	0
1947	0	0	0	1	1	0	1
1948	0	0	0	0	0	0	0
1949	0	0	0	0	1	1	0
1950	0	0	0	0	0	1	0
1951	0	0	0	0	0	1	1
1952	0	0	0	0	0	0	0
1953	0	0	0	0	0	0	1
1954	0	0	0	0	1	0	1
1955	0	0	0	0	0	0	0
1956	0	1	0	1	0	0	1
1957	0	0	1	0	0	0	1
1958	0	0	1	0	0	0	1
1959	0	0	0	0	0	0	1
1960	0	0	0	0	0	0	0
1961	0	0	0	1	0	0	0
1962	0	0	0	0	0	0	0
1963	0	0	0	0	0	1	0
1964	0	0	1	1	1	1	0
1965	0	0	0	0	0	0	1
1966	0	0	0	0	0	0	1
1967	0	0	0	0	1	1	1
1968	0	0	0	0	0	0	0
1969	0	0	0	0	0	0	0
1970	0	0	0	0	0	0	0
1971	0	0	0	0	0	0	0
1972	0	0	0	0	0	0	1
1973	0	0	0	0	0	0	0
1974	0	0	0	0	0	0	0
1975	0	0	0	0	0	1	0
1976	0	0	0	0	0	1	1
1977	0	0	0	0	0	1	0
1978	0	0	0	0	0	0	0
1979	0	0	0	1	1	1	1
1980	0	0	0	1	0	0	0
Outbreak Frequency	0	1	4	6	7	10	15

Table 1. Cell Values For The Seven Test Sites Of Historical Infestation In Wyoming (1944 – 1980)

Table 2 contains the hand simulated one-step transition probabilities that were calculated from the Markov analysis using the values from Table 1.

Site	**Probability**	
1	1.000	0.000
	#DIV/0!	#DIV/0!
2	0.971	0.029
	1.000	0.000
3	0.906	0.094
	0.750	0.250
4	0.839	0.161
	0.800	0.200
5	0.793	0.207
	0.857	0.143
6	0.808	0.192
	0.500	0.500
7	0.571	0.429
	0.600	0.400

Table 2. One-step Markov Chain Transition Probabilities* Derived From The Seven Test Sites Of Historical Infestation In Wyoming

According to Bhat [25] a state is called an absorbing state if and only if $P_{ii} = 1$; site 1 is an absorbing state. The one-step transition probabilities are the probability that next year will have a low or high density given that this year has low densities [34]. According to Table 2, if site 2 is currently uninfested, there is a 3% chance that it will be infested next year.

Table 3 contains the n-step transition probabilities that were calculated from the Markov analysis. From the one-step transition probabilities (see Table 2) the 2nd, 3rd, 4th and 5th step transition probabilities were calculated and steady state was achieved (see

---

* For each site the probabilities of transitions are shown as

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} \equiv \begin{bmatrix} \text{Uninfested to Uninfested} & \text{Uninfested to Infested} \\ \text{Infested to Uninfested} & \text{Infested to Infested} \end{bmatrix}$$

Table 3). For example according to Table 3, there is a 97% chance that site 2 will remain uninfested in 2, 3, 4, or 5 subsequent years if this year it is uninfested.

Site	2nd Subsequent Year		3rd Subsequent Year		4th Subsequent Year		5th Subsequent Year	
1	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
2	0.972	0.028	0.972	0.028	0.972	0.028	0.972	0.028
	0.971	0.029	0.972	0.028	0.972	0.028	0.972	0.028
3	0.892	0.108	0.889	0.111	0.889	0.111	0.889	0.111
	0.867	0.133	0.885	0.115	0.888	0.112	0.889	0.111
4	0.832	0.168	0.832	0.168	0.832	0.168	0.832	0.168
	0.831	0.169	0.832	0.168	0.832	0.168	0.832	0.168
5	0.806	0.194	0.806	0.194	0.806	0.194	0.806	0.194
	0.802	0.198	0.806	0.194	0.806	0.194	0.806	0.194
6	0.749	0.251	0.730	0.270	0.725	0.275	0.723	0.277
	0.654	0.346	0.701	0.299	0.716	0.284	0.720	0.280
7	0.584	0.416	0.583	0.417	0.583	0.417	0.583	0.417
	0.583	0.417	0.583	0.417	0.583	0.417	0.583	0.417

Table 3. Nth-Step Transition Probabilities Derived From The Seven Test Sites Of Historical Infestation In Wyoming

Table 4 shows the occupation time (expected length of time a cell will spend in a state before switching to the other state) and variance and Table 5 shows the 95% confidence interval that were calculated from the Markov analysis. Site 2 would experience low densities for 34 years before switching to high, with a variance of 1190 and a 95% confidence interval of 11.4.

Site	Expected length of time with low densities before switching to high	Variance	Expected length of time with high densities before switching to low	Variance
1	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
2	34.000	1190.000	0.000	0.000
3	9.667	103.111	0.333	0.444
4	5.200	32.240	0.250	0.313
5	3.833	18.528	0.167	0.194
6	4.200	21.840	1.000	2.000
7	1.333	3.111	0.667	1.111

Table 4. Occupation Time Derived From The Seven Test Sites Of Historical Infestation In Wyoming

Site	95% Confidence Interval for low densities before switching to high	95% Confidence Interval for high densities before switching to low
1	#DIV/0!	#DIV/0!
2	11.384	0.000
3	3.351	0.220
4	1.874	0.184
5	1.420	0.146
6	1.542	0.467
7	0.582	0.348

Table 5. Confidence Interval Derived From The 7 Test Sites Of Historical Infestation In Wyoming

Table 6 shows the expected number of years and percentage of time during the next 5 years each site will have low or high densities if they started with low or high densities. During the next 5 years, site 2 is expected to experience low densities if it started with low densities for 4.9 years or 97% of the time.

Site	Expected Length of time in 5 Years will experience low densities if started low	Expected Length of time in 5 Years will experience high densities if started low	Expected Length of time in 5 Years will experience low densities if started high	Expected Length of time in 5 Years will experience high densities if started high
	Number of Years			
	% of Time			
1				
	0%	0%	0%	0%
2	4.865	0.135	4.865	0.135
	97%	3%	97%	3%
3	4.459	0.541	4.459	0.541
	89%	11%	89%	11%
4	4.026	0.974	4.026	0.974
	81%	19%	81%	19%
5	4.054	0.946	4.054	0.942
	81%	19%	81%	19%
6	3.649	1.351	3.646	1.354
	73%	27%	73%	27%
7	2.973	2.027	2.973	2.027
	59%	41%	59%	41%

Table 6. Number Of Years And Percent Of 5 Years Each Location Will Experience Low Or High Densities If Started Low/High

Test of Model Output Values

To test the results of the prototype extension, the prototype was run on the 1944 to 1980 maps and the values were collected. The values were compared with the hand-simulated data from same seven sites (Tables 2-6). Table 7 contains the difference in the 1 to 5-step transition probabilities that were calculated by the hand simulation and the computed by the hand simulation and the prototype. It was shown that the results produced by the GIS were similar to the hand simulation. For example, the hand simulation computed that site 2 (see Table 2) had a 3% chance of going from uninfested to infested while the prototype computed that site 2 (see Table 7) had a 3% chance. As shown in Table 7, the prototype computed that site 2 had a 97% chance of remaining uninfested if started uninfested, similar to that of the hand simulation, (see Table 3). Site 2 would remain uninfested for 34 years in the hand simulation (see Table 4) and 34 years in the prototyped extension (see Table 8). In the next 5 years, if site 2 started uninfested, it would remain uninfested for 4.9 years according to the hand simulation (see Table 6) and 4.9 according to the prototype, as shown in Table 10. Only minor differences (see Table 10) can be seen between the hand simulation and the values produced by the prototype.

Site	Next Year		2nd Subsequent Year		3rd Subsequent Year		4th Subsequent Year		5th Subsequent Year	
1	0.000	0.000	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.009	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 7. Difference Between The Hand Simulation And The Prototype - 5 Years Of Transition Probabilities For The Seven Test Sites

Site	Expected length of time with low densities before switching to high	Variance	Expected length of time with high densities before switching to low	Variance
1	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000

Table 8. Difference Between The Hand Simulation And The Prototype - Occupation Time For The Seven Test Sites

Site	95% Confidence Interval for low densities before switching to high	95% Confidence Interval for high densities before switching to low
1	#DIV/0!	#DIV/0!
2	0.000	0.000
3	0.000	0.000
4	0.000	0.000
5	0.000	0.000
6	0.000	0.000
7	0.000	0.000

Table 9. Difference Between The Hand Simulation And The Prototype - Confidence Interval For The Seven Test Sites

Site	Expected Length of time in 5 Years will experience low densities if started low	Expected Length of time in 5 Years will experience high densities if started low	Expected Length of time in 5 Years will experience low densities if started high	Expected Length of time in 5 Years will experience high densities if started high
	Number of Years			
	% of Time			
1	0.000	0.000	0.000	0.000
	0%	0%	0%	0%
2	0.004	-0.004	0.004	-0.004
	0%	0%	0%	0%
3	0.015	-0.015	0.015	-0.015
	0%	0%	0%	0%
4	-0.135	0.135	-0.135	0.135
	-3%	3%	-3%	3%
5	0.026	-0.026	0.026	-0.030
	1%	-1%	1%	-1%
6	0.037	-0.037	0.037	-0.037
	1%	-1%	1%	-1%
7	0.056	-0.056	0.056	-0.056
	1%	-1%	1%	-1%

Table 10. Difference Between The Hand Simulation And The Prototype - Number Of Years And Percent Of 5 Years Each Location Will Experience Low Or High Densities If Started Low/High For The Seven Test Sites

Testing Against Real World Values

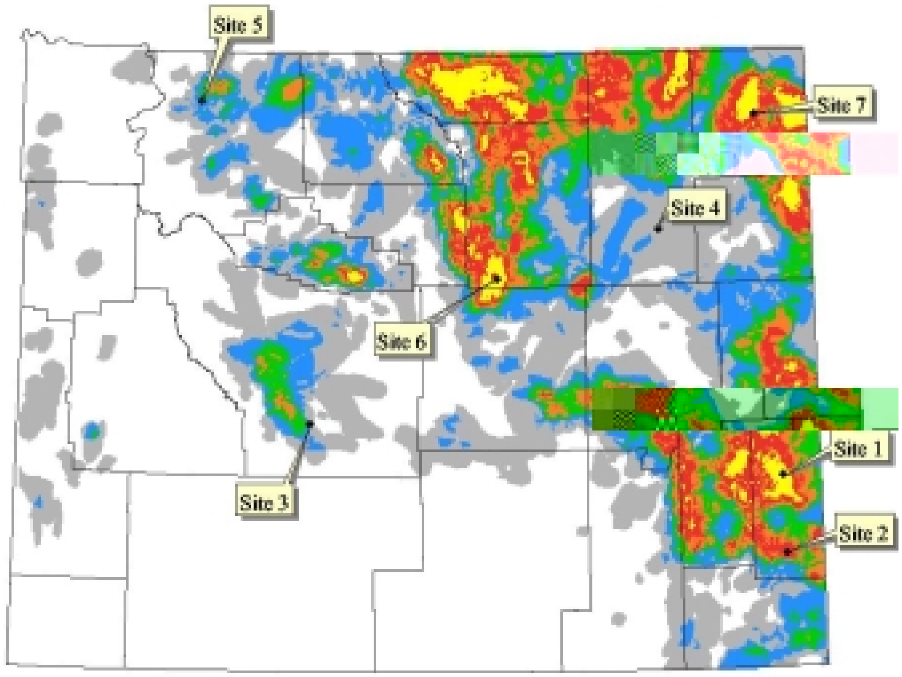


Figure 4. The Outbreak Frequency Map For Wyoming With The Seven Test Sites For Testing Cell Values During 1975 To 1980

Figure 4 shows the seven test sites to be used to test the real world values overlaid on the "Outbreak Frequency" map. To test the accuracy of the model against real data, the prototype was run on the maps for years 1944 to 1975 and seven sites were analyzed, as seen in Figure 4. The values of the maps were then compared to 1976 to 1980 for accuracy. Table 11 contains the cell values of the seven points from 1975 to 1980.

Site	1	2	3	4	5	6	7
Location / Year	261000,142000	264000,87000	-77000,178000	172000,319000	-154000,411000	57000,282000	240000,402000
75	0	1	1	0	1	1	0
76	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0
78	1	1	0	0	0	0	0
79	0	0	0	0	0	0	1
80	0	0	0	0	0	0	0

Table 11. 1975 - 1980 Cell Values For Seven Test Sites

Table 12 contains the 5th subsequent year Markov chain transition probabilities that were calculated by the prototype. For example, the prototype computed that site 2 had a 74% chance if currently infested of becoming uninfested in 5 subsequent years as seen in Table 12.

Site	Probabilities	
1	0.547	0.453
	0.550	0.450
2	0.741	0.259
	0.741	0.259
3	0.938	0.063
	0.938	0.062
4	0.937	0.063
	0.915	0.085
5	0.835	0.165
	0.835	0.165
6	0.596	0.404
	0.587	0.413
7	0.742	0.258
	0.742	0.258

Table 12. 5th Year Markov Chain Transition Probabilities Derived From The Seven Test Sites

Table 13 shows the expected number of years and percentage of time during the next 5 years each site will have low or high densities if they started with low or high densities. For example, site 2 if infested this year should remain uninfested for 3.7 years before switching to infested.

Site	Expected Length of time in 5 Years will experience low densities if started low	Expected Length of time in 5 Years will experience high densities if started low	Expected Length of time in 5 Years will experience low densities if started high	Expected Length of time in 5 Years will experience high densities if started high
	Number of Years			
	% of time			
1	2.741	2.259	2.743	2.257
	55%	45%	55%	45%
2	3.704	1.296	3.704	1.296
	74%	26%	74%	26%
3	4.688	0.313	4.688	0.312
	94%	6%	94%	6%
4	4.679	0.321	4.657	0.343
	94%	6%	93%	7%
5	4.175	0.825	4.175	0.825
	84%	16%	84%	16%
6	2.967	2.033	2.958	2.042
	59%	41%	59%	41%
7	3.710	1.290	3.710	1.290
	74%	26%	74%	26%

Table 13. Number Of Years And Percent Of 5 Years Each Location Will Experience Low Or High Densities If Started Low/High Derived By The Prototype For The Seven Test Sites

It was found that in 1975 (see Table 11), site 1 was considered uninfested and according to Table 13 it will spend 2.74 years uninfested until switching to infested. As shown in Table 11 site 1 was uninfested for 1976 and 1977 and then infested for 1978, therefore uninfested for 3 years. All of the sites except 6 behaved similar to the prediction. Site 6 was expected to be infested for two years and it was only infested 1 year.

Based on this evaluation, it appears that a GIS and a two-state Markov chain can be used to accurately predict grasshopper infestations. In the future this method may be applied to managing grasshopper infestations throughout the West.

Chapter 6

Comparisons and Summary

For this project a prototype extension was created for ArcView that applied a two-state Markov chain to grasshopper infestation maps in Wyoming to predict future population dynamics. The maps gathered by Schell et al. [9] were converted from ERDAS format to a grid format that ArcView could use. Then using a prototyping method, the two-state Markov chain extension was written using Avenue. The extension added a new menu choice to ArcView that allowed the user to perform five functions. The function "Outbreak Frequency" computed the number of times each cell was infested. "One Year Transition Probabilities" computed the one-step Markov chain transition probabilities. The choice "Expected Number of Years" computed the occupation time, variance and 95% confidence interval. "Probabilities in 2, 3, 4, 5 Future Years" computed the n-step Markov chain transition probabilities, and "Percent of Years in 5 in each Transition" computed the number of years within 5 years and percent of 5 years each cell is expected to be in each state before switching to the other state.

The extension was designed so that the user could load and unload the extension, as needed. To run the extension, the user opens a view with more than 1 grid present. Three of the menu choices are not available until the user performs the one-year transition probabilities. For each of the menu choices, a new view is created, and the information displayed as a grid.

To verify the results of the prototype, seven sites were chosen and hand simulation was completed. It was found that the hand simulation values and the prototype's results were similar.

To verify the results of the prototype with real world values, seven sites were chosen, the prototype was run on the 1944 to 1975 data. The predictions were compared against the 1976 to 1980 data to see how the results compared with real world values. As seen in Table 13 the prototype predicted: if site 1 started uninfested and should have

The original maps being based on categorical data (0-3, 4-7, 8-15, and >15), 4) Not knowing where and when grasshopper infestations were treated, and 5) Not knowing when grasshoppers died due to environmental factors. Still, it does not appear that these differences would significantly impact a landowner/manager's decisions about treating grasshopper infestations.

Some entomologists feel that landowner/managers can receive multiyear benefit from spraying insecticide. The Markov chain was applied to the maps from 1944 to 1996, and it was found that most locations in Wyoming tend to become or stay uninfested. Table 14 contains the percentage breakdowns, based on the Markov transition probabilities, of land that will remain uninfested or become uninfested if the land is currently uninfested or infested. For example, if Wyoming were currently uninfested there is a 90 to 100% chance that 87% of Wyoming would remain uninfested, however, if the state were, infested there is a 90 to 100% chance that 28% will become uninfested.

Uninfested to Uninfested	
Transition Probability	Percent of Land
0	0.0%
0-10	0.0%
10-20	0.0%
20-30	0.0%
30-40	0.0%
40-50	0.0%
50-60	0.0%
60-70	0.3%
70-80	2.1%
80-90	10.9%
90-100	86.7%
Infested to Uninfested	
0	0.0%
0-10	0.0%
10-20	0.0%
20-30	0.1%
30-40	0.5%
40-50	3.9%
50-60	1.6%
60-70	7.5%
70-80	4.9%
80-90	4.5%
90-100	27.8%
No Documented Infestation	49.2%

Table 14. Percentage Of Land In Wyoming That Will Move From One State To Another Or Remain The Same

Table 15 shows the percentage of and duration of time that is expected to remain uninfested or infested. A majority of Wyoming, if currently infested, can expect the infestation to last approximately one year as seen in Table 15. Approximately 50% of Wyoming has no documented infestation.

Future Years Uninfested (FYU)	
Years	**Percent of Land**
0-5	5.3%
5-10	13.1%
10-15	7.5%
15-20	0.1%
20-25	9.9%
25-30	0.0%
30-35	0.0%
35-40	0.0%
40-45	1.7%
45-50	13.2%
No Documented Infestation	49.2%

Future Years Infested (FYI)	
Years	**Percent of Land**
0-1	49.3%
1-2	1.3%
2-3	0.2%
3-4	0.0%
4-5	0.0%
No Documented Infestation	49.2%

Table 15. Percent Of Wyoming That Is Expected To Remain Uninfested Or Infested

Based on this work, it would seem that a vast majority of the landowners/managers are not going to receive multiyear benefit. Therefore, in deciding to treat their grasshopper infestation most of the landowners/managers need to base their cost:benefit on the current year, only.

Future work will involve comparing the Markov transition probability maps created with ecological information like vegetation, precipitation, elevation, evapotranspiration, landform, and soil associations.

Bibliography

1. Tappan, G.G., D.G. Moore, and W.I. Knausenberger, *Monitoring Grasshopper and Locust habitats in Sahelian Africa using GIS and Remote Sensing Technology*. International Journal of Geographical Information Systems, 1991. **5**(1): p. 123-135.
2. Schell, S.P. and J.A. Lockwood, *Spatial Analysis Optimizes Grasshopper Management.* GIS World, 1995(November): p. 68-73.
3. DeBrey, L.D., M.J. Brewer, and J.A. Lockwood, *Rangeland Grasshopper Management*, . 1993, University of Wyoming: Laramie.
4. Dysart, R.J., *Relative Importance of Rangeland Grasshoppers in Western North America: A Numerical Ranking From the Literature*, , United States Department of Agriculture Animal and Plant Health Inspection Service.
5. Hewitt, G.B. and J.A. Onsager, *Control of Grasshoppers on Rangeland in the United States -- A Perspective.* J. Range Manage., 1983. **36**(2): p. 202-207.
6. Lockwood, J.A., University of Wyoming, Personal Communication, 1999.
7. Pfadt, R.E., *Some Aspects of the Ecology of Grasshopper Populations Inhabiting the Shortgrass Plains*, in *Insect Ecology*. 1977, Univ. of MN. p. 107.
8. Blickenstaff, C.C., F.E. Skoog, and R.J. Daum, *Long-term Control of Grasshoppers.* Journal of Economic Entomology, 1974. **67**(2): p. 268-274.
9. Schell, S.P. and J.A. Lockwood, *Spatial Characteristics of Rangeland Grasshopper (Orthoptera: Acrididae) Populations Dynamics in Wyoming: Implications for Pest Management.* Environmental Entomology, 1997. **26**(5): p. 1056-1065.
10. Stewart, W.J., *Introduction to the Numerical Solution of Markov Chains*. 1994, Princeton: Princeton University Press. 539.
11. Chou, Y.-H., *Exploring Spatial Analysis in Geographic Information Systems*. First ed. 1997, Santa Fe: OnWord Press. 474.
12. Chrisman, N., *Exploring Geographic Information Systems*. 1997: John Wiley & Sons, Inc. 298.
13. Zesheng, Z. and S. Ling. *Design and Implementation of a Farm Weed Management System Based on GIS*. in *1997 ESRI User Conference*. 1997. San Diego, California.
14. Swetnam, R.D., *et al.*, *A Geographic Information System for Predicting Weed Changes on Set-Aside Arable Land.* Weed Technology, 1998. **12**: p. 53-63.
15. Gordon, D. *Vineyard and Winery Management: A Case Study in GIS Implementation*. in *1997 ESRI User Conference*. 1997. San Diego, California.
16. Mangold, G., *Precision Farming Modernizes Conventional Techniques.* GIS World, 1998. **11**(2): p. 48-51.
17. Abbors, S., R. Ghezelbash, and M. Jackson. *Using GIS Applications in Range Site Analysis*. in *1997 ESRI User Conference*. 1997. San Diego, California.
18. Walker, R. and L. Craighead. *Analyzing Wildlife Movement Corridors in Montana Using GIS*. in *1997 ESRI User Conference*. 1997. San Diego, California.

19. Swantek, P.J., W.L. Halvorson, and C.R. Schwalbe. *The Use of GIS to Study Fire History in Vegetation Communities in Southeastern Arizona*. in *1997 ESRI User Conference*. 1997. San Diego, California.
20. Bryceson, K.P., *The use of Landsat MSS data to determine the distribution of locust eggbeds in the Riverina region of New South Wales, Australia.* International Journal Remote Sensing, 1989. **10**(11): p. 1749-1762.
21. Fielding, D.J. and M.A. Brusven, *Spatial analysis of grasshopper density and ecological disturbance on southern Idaho rangeland.* Agriculture, Ecosystems and Environment, 1993. **43**: p. 31-47.
22. Burckley, M., *Grasshoppers Meet GIS.* Quantum, 1995. **12**(1): p. 14-15.
23. Cigliano, M.M., W.P. Kemp, and T.M. Kalaris, *Spatiotemporal Characteristics of Rangeland Grasshopper (Orthoptera: Acrididae) Regional Outbreaks in Montana.* Journal of Orthoptera Research, 1995. **4**: p. 111-126.
24. Schell, S.P. and J.A. Lockwood, *Spatial Analysis of Ecological Factors Related to Rangeland Grasshopper (Orthoptera: Acrididae) Outbreaks in Wyoming.* Environmental Entomology, 1997. **26**(6): p. 1343-1353.
25. Bhat, U.N., *Elements of Applied Stochastic Processes*. Second ed. 1984, New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons. 684.
26. Stander, J., *et al.*, *Markov Chain Analysis and Specialization in Criminal Careers.* The British Journal of Criminology, 1989. **29**(4): p. 317-335.
27. Montgomery, B.A., J.W. Lloyd, and A. Jensen, *Use of a Markov-Chain Model to Evaluate Employment Policies for a Veterinary School Faculty.* Journal of Veterinary Medical Education, 1994. **21**(1): p. 28-32.
28. Pankin, M., *Markov Models/Batting Order Optimization*, www.retrosheet.com/mdp/markov, 1997.
29. Berentsen, P.B.M., A.A. Dijkhuizen, and A.J. Oskam, *A Dynamic Model for Cost-Benefit Analyses of Foot-And-Mouth Disease Control Strategies.* Preventive Veterinary Medicine, 1992. **12**: p. 229-243.
30. Carpenter, T.E., *Microcomputer Programs for Markov and Modified Markov Chain Disease Models.* Preventive Veterinary Medicine, 1988. **5**: p. 169-179.
31. Rosen, L. and G. Gustafson, *A Bayesian Markov Geostatistical Model for Estimation of Hydrogeological Properties.* Ground Water, 1996. **34**(5): p. 865-875.
32. Kim, C.S., W. Lin, and M.N. Leath, *The Changing Structure of the U.S. Flour Milling Industry.* The Journal of Agricultural Economics Research, 1991. **43**(3): p. 18-25.
33. Mukerji, M.K. and H.N. Hayhoe, *Probability Analysis of Fluctuations of Grasshopper Populations in Saskatchewan.* The Canadian Entomologist, 1988. **120**(December): p. 1063-1070.
34. Kemp, W.P., *Probability of Outbreak for Rangeland Grasshoppers (Orthoptera: Acrododae) in Montana: Application of Markovian Principles.* Journal of Economic Entomology, 1987. **80**(6): p. 1100-1105.
35. Lockwood, J.A. and W.P. Kemp, *Probabilities of Rangeland Grasshopper Outbreaks in Wyoming Counties*, 1987, University of Wyoming: Laramie. p. 17.

36. Davis, R.M., *et al.*, *The Economic Threshold for Grasshopper Control on Public Rangelands.* Journal of Agricultural and Resource Economics, 1992. **17**(1): p. 56-65.
37. Lockwood, J.A., University of Wyoming, Personal Communication, 1998.
38. Jeffers, J.N.R., *An Introduction to Systems Analysis: with Ecological Applications*. Contemporary Biology, ed. E.J.W. Barrington, A.J. Willis, and M.A. Sleigh. 1978, London: Edward Arnold Limited. 198.
39. Bhat, U.N., *Elements of Applied Stochastic Processes*. 1972: John Wiley & Sons, Inc.
40. Agnew, J.L. and R.C. Knapp, *Linear Algebra with Applications*. Third ed. 1978, Pacific Grove: Brooks/Cole Publishing Company. 392.
41. Pfleeger, S.L., *Software Engineering The Production of Quality Software*. Second ed. 1991, New York: Macmillan Publishing Company. 516.
42. Berry, J.S., *et al.*, *Assessing Rangeland Grasshopper Populations*, , United States Department of Agriculture Animal and Plant Health Inspection Service.
43. Schell, S.P., *Spatial Analysis of Ecological Factors Related to Grasshopper (Orthoptera: Acrididae) Population Dynamics in Wyoming*, in *Department of Plant, Soil, and Insect Sciences*. 1994, University of Wyoming: Laramie. p. 127.

Appendix A

'Name: Markov.CreateLegends

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will create legends for the different themes given a legend file name
'
'Requires:
'    Calls:
'
'Self: (theme legend, legend filename, legend load type enum)
'Returns:
'-----------------------------------------------------------------------

themelegend = self.Get(0)
legendtoapply = self.Get(1)
LoadTypeEnum = self.Get(2)

returnbool = themelegend.Load(legendtoapply.asfilename,LoadTypeEnum)  'Boolean
        represents a successful or unsuccessful application of the legend
If (returnbool.not) then
  msgbox.info("Legend File Is Missing!","Error")   'Legend ramp file is not available
        therefore display and error message
end
```

'Name: Markov.FutProb

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will compute the probability of finding densities in future years.
'For n:
'  P00(n) = P00^(n-1)*P00 + P01^(n-1)*P10
'  P01(n) = P00^(n-1)*P01 + P01^(n-1)*P11
'  P10(n) = P10^(n-1)*P00 + P11^(n-1)*P10
'  P11(n) = P10^(n-1)*P01 + P11^(n-1)*P11
'
'Requires:
'  Calls:  Markov.CreateLegends
'
'Self:(P00,P01,P10,P11,P00(n-1),P02(n-1),P10(n-1),P11(n-1),View,Year)
'Returns:
'-----------------------------------------------------------------------

P00Grid = self.Get(0)
P01Grid = self.Get(1)
P10Grid = self.Get(2)
```

```
P11Grid = self.Get(3)
NP00Grid = self.Get(4)
NP01Grid = self.Get(5)
NP10Grid = self.Get(6)
NP11Grid = self.Get(7)
SeqYearsView = self.Get(8)
y = self.get(9)

NewP00 = (NP00Grid * P00Grid) + (NP01Grid * P10Grid)      'Computes the new
        transition probability grids
NewP01 = (NP00Grid * P01Grid) + (NP01Grid * P11Grid)
NewP10 = (NP10Grid * P00Grid) + (NP11Grid * P10Grid)
NewP11 = (NP10Grid * P01Grid) + (NP11Grid * P11Grid)

NewP00theme = GTheme.Make(NewP00)             'create grid themes for the computed
        grids
NewP01theme = GTheme.Make(NewP01)
NewP10theme = GTheme.Make(NewP10)
NewP11theme = GTheme.Make(NewP11)

SeqYearsView.AddTheme(NewP00theme)     'Add the maps showing the future
        transition probability to the view
SeqYearsView.AddTheme(NewP01theme)
SeqYearsView.AddTheme(NewP10theme)
SeqYearsView.AddTheme(NewP11theme)

NewP00themelegend = NewP00theme.GetLegend    'Get the current legend and change it
        to a predefined legend
NewP01themelegend = NewP01theme.GetLegend
NewP10themelegend = NewP10theme.GetLegend
NewP11themelegend = NewP11theme.GetLegend
results =
        av.run("Markov.CreateLegends",{NewP00themelegend,"tranprobb1.avl",#LEGE
        ND_LOADTYPE_ALL})
results =
        av.run("Markov.CreateLegends",{NewP01themelegend,"tranproba1.avl",#LEGE
        ND_LOADTYPE_ALL})
results =
        av.run("Markov.CreateLegends",{NewP10themelegend,"tranprobb1.avl",#LEGE
        ND_LOADTYPE_ALL})
results =
        av.run("Markov.CreateLegends",{NewP11themelegend,"tranproba1.avl",#LEGE
        ND_LOADTYPE_ALL})

If (y = 2) then
```

```
  NewP00theme.SetVisible(True)        'turn on the themes with the future transition
          probabilities for the 2nd year
  NewP01theme.SetVisible(True)
  NewP10theme.SetVisible(True)
  NewP11theme.SetVisible(True)
end

If (y = 2) then                       'Create the 1st part of the theme title to represent the year
          for the transition
  years = "2nd"
  ElseIf (y = 3) then
    years = "3rd"
    Elseif (y = 4) then
      years = "4th"
    else
      years = "5th"
End

NewP00theme.SetName(years+" Year Transition From Uninfested to Uninfested")
          'Create a theme name that will represent the grid information
NewP01theme.SetName(years+" Year Transition From Uninfested to Infested")
NewP10theme.SetName(years+" Year Transition From Infested to Uninfested")
NewP11theme.SetName(years+" Year Transition From Infested to Infested")
```

'Name: Markov.GetPN-1

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will get the transition probability grids for n-1
'
'Requires:
'  Calls:  Markov.FutProb
'
'Self:(P00,P01,P10,P11,View,Year)
'Returns:
'-----------------------------------------------------------------------

P00Grid = self.Get(0)
P01Grid = self.Get(1)
P10Grid = self.Get(2)
P11Grid = self.Get(3)
SeqYearsView = self.Get(4)
y = self.get(5)

ny = y - 1                      'need the previous year
```

```
'Create the 1st part of the theme title to represent the year for the transition
If (ny = 2) then
  years = "2nd"
  ElseIf (ny = 3) then
    years = "3rd"
    Elseif (ny = 4) then
      years = "4th"
    else
      years = "5th"
End

'Find the grids in the Future year transition probabilities for n-1
NP00 = SeqYearsView.FindTheme(years+" Year Transition From Uninfested to
        Uninfested")
NP00Grid = NP00.GetGrid
NP01 = SeqYearsView.FindTheme(years+" Year Transition From Uninfested to
        Infested")
NP01Grid = NP01.GetGrid
NP10 = SeqYearsView.FindTheme(years+" Year Transition From Infested to
        Uninfested")
NP10Grid = NP10.GetGrid
NP11 = SeqYearsView.FindTheme(years+" Year Transition From Infested to Infested")
NP11Grid = NP11.GetGrid

results =
        av.run("Markov.FutProb",{P00Grid,P01Grid,P10Grid,P11Grid,NP00Grid,NP01G
        rid,NP10Grid,NP11Grid,SeqYearsView,y})  'calls the script to calculate the
        future probabilities
```

'Name: Markov.NumYears

```
'
'Title: Computes the Number of Expected Years at each State
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will compute the expected number of years a location will experience a state
        before switching to the other state
'will create a view called "Expected Number of Future Years" to display the expected 2nd.
        year maps, 2 variance maps, and the 95% Confidence Interval
'E0 = (1-a)/a (Expected length of time with low densities before switching to high
        densities)
'E1 = (1-b)/b (Expected length of time with high densities before switching to low)
'V0 = (1-a)/a2 (variance for E0)
'V1 = (1-b)/b2 (variance for E1)
```

```
'95%CI = 0.33 * SQRT(Variance)
'
'Requires: View - that contains the 1-year transition probabilities named "1 Year
        Transition Probabilities"
'         Spatial Analyst
'         Legend Files: varianceb,confint,expyearsb,variancea,convintb,expyearsa
'
'  Calls:  Markov.SetUpGrids
'           Markov.SetProj
'
'Self:
'Returns:
'----------------------------------------------------------------------

theProject = av.GetProject
TranProbView = theProject.FindDoc("1 Year Transition Probabilities")   'returns the
        view named "1 Year Transition Probabilities"

If (TranProbView = nil) then                              'If the needed view is missing error
        message otherwise compute the expected number of years and variances
  msgbox.info("Missing the View called 1 Year Transition Probabilities","Error")
else
  TranProbViewProjection = TranProbView.GetProjection         'returns the projection of
        the view containing the transition probabilities
  DataViewExtent = TranProbView.ReturnExtent

  av.GetProject.GetWorkDir.SetCwd                     'get the current working directory
  a = TranProbView.FindTheme("Transition From Uninfested to Infested")  'get the grids
        that represent the transitions from 0 to 1 and 1 to 0
  aGrid = a.GetGrid
  b = TranProbView.FindTheme("Transition From Infested to Uninfested")
  bGrid = b.GetGrid
  E0 = (1.AsGrid - aGrid)/aGrid                'E0 = (1-a)/a  and E1 = (1-b)/b
  E1 = (1.AsGrid - bGrid)/bGrid

'E0 = E0 - 1.AsGrid                            'Subtract off current year
'E1 = E1 - 1.AsGrid

  V0 = (1.AsGrid - aGrid)/(aGrid^2.AsGrid)               'V0 = (1-a)/a2  and V1 = (1-b)/b2
  V1 = (1.AsGrid - bGrid)/(bGrid^2.AsGrid)

  CI0 = 0.33.AsGrid * V0.Sqrt               '95% Confidence Interval
  CI1 = 0.33.AsGrid * V1.Sqrt

  ExpYearsView = View.Make         'Create a view that will display the expected number
        of years maps
```

```
  results = av.run("Markov.SetUpGrids",{ExpYearsView,V0,False,"Variance in
          FYU","varianceb.avl","V0theme"})  'calls the script to set the project of the view
  results = av.run("Markov.SetUpGrids",{ExpYearsView,CI0,True,"95% Confidence
          Interval for FYU","confint.avl","CI0theme"})
  results = av.run("Markov.SetUpGrids",{ExpYearsView,E0,True,"Future Years
          Uninfested (FYU)","expyearsb.avl","E0theme"})
  results = av.run("Markov.SetUpGrids",{ExpYearsView,V1,False,"Variance in
          FYI","variancea.avl","V1theme"})
  results = av.run("Markov.SetUpGrids",{ExpYearsView,CI1,True,"95% Confidence
          Interval for FYI","confintb.avl","CI1theme"})
  results = av.run("Markov.SetUpGrids",{ExpYearsView,E1,True,"Future Years Infested
          (FYI)","expyearsa.avl","E1theme"})


  ExpYearsViewProjection = ExpYearsView.GetProjection  'Returns the projection of the
          new view (Expected number of years view)
  results = av.run("Markov.SetProj",{ExpYearsViewProjection,ExpYearsView})  'calls
          the script to set the project of the view
  ExpYearsViewDisplay = ExpYearsView.GetDisplay      'get the display of the new view
  ExpYearsViewDisplay.ZoomToRect(DataViewExtent)      'Zoom to the same location at
          the data view

  ExpYearsView.SetComments("Expected Length of Time, Variances, and
          95%Confidence Interval at 0 or 1 before switching")    'Set up the views comment
          field
  ExpYearsView.SetName("Expected Number of Years")    'Set the views name to
          represent the maps displayed
  ExpYearsWindow = ExpYearsView.GetWin
  ExpYearsWindow.Open
end
```

'Name: Markov.NumYearsToolUpdate

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will enable the menu options if the correct view exists.
'
'Requires: A View called 1 Year Transition Probabilities
'         Spatial Analyst
'
'   Calls:
'
'Self:
'Returns:
```

```
else
 GridList = List.Make                 'Create any empty list that will contain the list of
        themes that are not Grid themes
 for each i in 0..(ThemeList.count - 1)
  If (ThemeList.Get(i).Is(GTheme)) then      'Create a list that contains only the grid
        themes
    GridList = GridList.Add(ThemeList.Get(i))
  end                                'end if themelist is a GTHeme
 end                                'end for each themelist

 If (GridList.Count = 0) then          'No grid themes display an error message
   msgbox.info("There are no grid themes in the view, Please add some and
        rerun.","Error")
 Else
   theProjection = DataView.GetProjection
   DataViewExtent = DataView.ReturnExtent
   n = GridList.Count               'n (number of grids)
   tempgrid = Grid.MakeFromNumb(0)       'create temp. grids that will start with 0 then
        contain the number of times a cell in infested
   for each i in 0..(n-1)
     g1 = GridList.Get(i)
     g1Grid = g1.GetGrid
     tempgrid = tempgrid + g1grid      'incrementally sum up all the grids in the view
   end

   OutFreqView = View.Make           'Create a view that will display the outbreak
        frequency

   results = av.run("Markov.SetUpGrids",{OutFreqView,tempgrid,True,"Outbreak
        Frequency","outfreq.avl","tempgridtheme"})  'calls the script to create a GTheme
        and add it to the view and set up its legend


   OutFreqViewProjection = OutFreqView.GetProjection          'returns the projection of
        the new view (outbreak frequency)
   results = av.run("Markov.SetProj",{OutFreqViewProjection,OutFreqView})  'calls the
        script to set the project of the view
   OutFreqViewDisplay = OutFreqView.GetDisplay  'Set the display of the outbreak
        frequency to be the same as the data view
   OutFreqViewDisplay.ZoomToRect(DataViewExtent)    'Zoom in on the area of
        interest

   OutFreqView.SetName("Outbreak Frequency Map")    'Set the Name of the view that
        contains the outbreak frequency
   OutFreqWindow = OutFreqView.GetWin
   OutFreqWindow.Open                            'Opens the new view window
```

```
  end                              'if gridlist.count = 0
end                                'If ThemeList.count = 0
```

'Name: Markov.Partof5

```
'
'Title: Computes next n year's infestation
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will create 4 grids that will represent the number of years in the next 5 years
        each transition will occur
'and the percentage of time before each transition.
'
'U00 = Summation (k = 1 to n)((b/(a+b)) + ((a(1-a-b)^k)/a+b))
'U01 = Summation (k = 1 to n)((a/(a+b)) - ((a(1-a-b)^k)/a+b))
'U10 = Summation (k = 1 to n)((b/(a+b)) - ((b(1-a-b)^k)/a+b))
'U11 = Summation (k = 1 to n)((a/(a+b)) + ((b(1-a-b)^k)/a+b))
' percentages of year grids are Uxx/n
'
'Requires: A View document called "Future Year Transition Probabilities" with the 5th
        year transition probabilities
'       Spatial Analyst
'       legend files: ua,purfutysa,ub,purfutysb
'
'   Calls:  Markov.U00
'           Markov.U01
'           Markov.U10
'           Markov.U11
'           Markov.SetUpGrids
'           Markov.SetProj
'
'Self:
'Returns:
'-----------------------------------------------------------------------

theProject = av.GetProject
FutProbView = theProject.FindDoc("Future Years Transition Probabilities")

If (FutProbView = nil) then                    'If the needed view is missing error
        message otherwise compute the number of years and percentages
  msgbox.info("Missing the View called Future Years Transition Probabilities","Error")
else
  FutProbViewProjection = FutProbView.GetProjection        'returns the projection of
        the view containing the transition probabilities
```

```
DataViewExtent = FutProbView.ReturnExtent

av.GetProject.GetWorkDir.SetCwd                    'get the current working directory
a = FutProbView.FindTheme("5th Year Transition From Uninfested to Infested")  'get
        the grids that represent the transitions from 0 to 1 and 1 to 0
aGrid = a.GetGrid
b = FutProbView.FindTheme("5th Year Transition From Infested to Uninfested")
bGrid = b.GetGrid

'calls the script to compute the U formulas
U00 = (av.run("Markov.U00",{aGrid,bGrid,1.AsGrid})) +
        (av.run("Markov.U00",{aGrid,bGrid,2.AsGrid})) +
        (av.run("Markov.U00",{aGrid,bGrid,3.AsGrid})) +
        (av.run("Markov.U00",{aGrid,bGrid,4.AsGrid}))  +
        (av.run("Markov.U00",{aGrid,bGrid,5.AsGrid}))
U01 = (av.run("Markov.U01",{aGrid,bGrid,1.AsGrid})) +
        (av.run("Markov.U01",{aGrid,bGrid,2.AsGrid})) +
        (av.run("Markov.U01",{aGrid,bGrid,3.AsGrid})) +
        (av.run("Markov.U01",{aGrid,bGrid,4.AsGrid}))  +
        (av.run("Markov.U01",{aGrid,bGrid,5.AsGrid}))
U10 = (av.run("Markov.U10",{aGrid,bGrid,1.AsGrid})) +
        (av.run("Markov.U10",{aGrid,bGrid,2.AsGrid})) +
        (av.run("Markov.U10",{aGrid,bGrid,3.AsGrid})) +
        (av.run("Markov.U10",{aGrid,bGrid,4.AsGrid}))  +
        (av.run("Markov.U10",{aGrid,bGrid,5.AsGrid}))
U11 = (av.run("Markov.U11",{aGrid,bGrid,1.AsGrid})) +
        (av.run("Markov.U11",{aGrid,bGrid,2.AsGrid})) +
        (av.run("Markov.U11",{aGrid,bGrid,3.AsGrid})) +
        (av.run("Markov.U11",{aGrid,bGrid,4.AsGrid}))  +
        (av.run("Markov.U11",{aGrid,bGrid,5.AsGrid}))

'compute the percentage by Uxx/n
n = 5
P00 = U00/n
P01 = U01/n
P10 = U10/n
P11 = U11/n

NFutYearsView = View.Make          'Create a view that will display the expected
        number of years maps

'Calls the script that will set up the grids and the legends
results = av.run("Markov.SetUpGrids",{NFutYearsView,U00,True,"Future Years at
        Uninfested if Started at Uninfested","ua.avl","U00theme"})  'calls the script to set
        the project of the view
```

```
  results = av.run("Markov.SetUpGrids",{NFutYearsView,P00,True,"Percentage of Time
         in 5 years Uninfested Starting Uninfested","purfutysa.avl","P00theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,U01,True,"Future Years at
         Infested if Started at Uninfested","ub.avl","U01theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,P01,True,"Percentage of Time
         in 5 years Infested Starting Uninfested","purfutysb.avl","P01theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,U10,True,"Future Years at
         Uninfested if Started at Infested","ua.avl","U10theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,P10,True,"Percentage of Time
         in 5 years Uninfested Starting Infested","purfutysb1.avl","P10theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,U11,True,"Future Years at
         Infested if Started at Infested","ub.avl","U11theme"})
  results = av.run("Markov.SetUpGrids",{NFutYearsView,P11,True,"Percentage of Time
         in 5 years Infested Starting Infested","purfutysa1.avl","P11theme"})

  NFutYearsViewProjection = NFutYearsView.GetProjection  'Returns the projection of
         the new view (Expected number of years view)
  results = av.run("Markov.SetProj",{NFutYearsViewProjection,NFutYearsView})  'calls
         the script to set the project of the view
  NFutYearsViewDisplay = NFutYearsView.GetDisplay      'get the display of the new
         view
  NFutYearsViewDisplay.ZoomToRect(DataViewExtent)      'Zoom to the same location
         at the data view

  NFutYearsView.SetComments("Number of years and percentage of time in each state
         that are expected during a 5 year period")    'Set up the views comment field
  NFutYearsView.SetName("Expected Number of Future Years at Each State")    'Set the
         views name to represent the maps displayed
  NFutYearsWindow = NFutYearsView.GetWin
  NFutYearsWindow.Open
end
```

'Name: Markov.Partof5ToolUpdate

```
'
'Author: Kiana Zimmerman
'---------------------------------------------------------------------
'This Script will enable the menu options if the correct view exists.
 '
'Requires: A View called “Future Years Transition Probabilities”
'         spatial analyst
'
'   Calls:
'
'Self:
'Returns:
```

```
'----------------------------------------------------------------------

theProject = av.GetProject
enabled = FALSE

TheView = theProject.FindDoc("Future Years Transition Probabilities")

If (TheView = nil) then
  enabled = False
else
  enabled = TRUE
end

SELF.SetEnabled(enabled)
```

'Name: Markov.ProbFutureYears

```
'
'Title: Compute the probabilities in 2, 3, 4, & 5 future years
'
'Author: Kiana Zimmerman
'----------------------------------------------------------------------
'This Script will compute the probability of finding densities in future years.
'
'Requires: A view called "1 Year Transition Probabilities" with 4 grids named
        "Uninfested to Uninfested", "Uninfested to Infested",
'          "Infested to Uninfested", and "Infested to Infested"
'       Spatial Analyst
'  Calls:  Markov.FutProb
'       Markov.GetPN-1
'       Markov.SetProj
'
'Self:
'Returns:
'----------------------------------------------------------------------

theProject = av.GetProject

TransView = theProject.FindDoc("1 Year Transition Probabilities")   'returns the view
        named "1 Year Transition Probabilities"

If (TransView = nil) then                    'If the needed view is missing error
        message otherwise compute future probabilities
  msgbox.info("Missing the View called 1 Year Transition Probabilities","Error")
else
  TransViewProjection = TransView.GetProjection
```

```
  DataViewExtent = TransView.ReturnExtent

  P00 = TransView.FindTheme("Transition From Uninfested to Uninfested")    'Find the
          grids in the "1 Year Transition" View
  P00Grid = P00.GetGrid
  P01 = TransView.FindTheme("Transition From Uninfested to Infested")
  P01Grid = P01.GetGrid
  P10 = TransView.FindTheme("Transition From Infested to Uninfested")
  P10Grid = P10.GetGrid
  P11 = TransView.FindTheme("Transition From Infested to Infested")
  P11Grid = P11.GetGrid

  SeqYearsView = View.Make         'Create a view that will display the expected
          probability in years maps

  results =
          av.run("Markov.FutProb",{P00Grid,P01Grid,P10Grid,P11Grid,P00Grid,P01Grid,
          P10Grid,P11Grid,SeqYearsView,2})  'calls the script to calculate the future
          probabilities
  results = av.run("Markov.GetPN-
          1",{P00Grid,P01Grid,P10Grid,P11Grid,SeqYearsView,3})  'calls the script to
          calculate the future probabilities
  results = av.run("Markov.GetPN-
          1",{P00Grid,P01Grid,P10Grid,P11Grid,SeqYearsView,4})
  results = av.run("Markov.GetPN-
          1",{P00Grid,P01Grid,P10Grid,P11Grid,SeqYearsView,5})

  SeqYearsViewProjection = SeqYearsView.GetProjection
  results = av.run("Markov.SetProj",{SeqYearsViewProjection,SeqYearsView})
  SeqViewYearsDisplay = SeqYearsView.GetDisplay
  SeqViewYearsDisplay.ZoomToRect(DataViewExtent)

  SeqYearsView.SetComments("nth Step Markov Transition probabilities.  (Shows the 0-
          >0, 0->1, 1->0, and 1->1 transitions in 2,3,4, or 5 subsequent years)")    'Set up
          the views comment field
  SeqYearsView.SetName("Future Years Transition Probabilities")    'Set the Name of the
          view that contains the transitions probabilities
  SeqYearsWindow = SeqYearsView.GetWin
  SeqYearsWindow.Open
end
```

'Name: Markov.ProbFutureYearsToolUpdate

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
```

```
'This Script will enable the menu options if the correct view exists.
 '
'Requires: A View called “1 Year Transition Probabilities”
'         Spatial Analyst
'
'    Calls:
'
'Self:
'Returns:
'-----------------------------------------------------------------------

theProject = av.GetProject
enabled = FALSE

TheView = theProject.FindDoc("1 Year Transition Probabilities")

If (TheView = nil) then
  enabled = False
else
  enabled = TRUE
end

SELF.SetEnabled(enabled)
```

'Name: Markov.ProbTrans

```
'
'Title: Computes 1st. year transition probabilities
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will create 4 grids that will represent the probability transitions based off
          grids in the data view
'The new grids will be displayed in a new view called "1 Year Transition Probabilities".
'
'The system has 2 possible states 0 (uninfested) and 1 (infested).  4 possible transitions
          from year to year 0->0, 0->1, 1->0, and 1->1
'Determine the proportion of time the system moved from one state to another and
          display in grid format these values.
'
'Requires: A View document with at least 2 grid themes  (assumes the grid themes are
          named in ascending order i.e. grid1 for grid at time 1, grid2 for the grid at time 2)
'         spatial analyst
'         legend files: tranproba.avl, tranprobb.avl, tranprobb1.avl, and tranproba1.avl
'
'    Calls:  Markov.SortGrid
```

```
'          Markov.SetProj
'          Markov.SetUpGrids
'
'Self:
'Returns:
'------------------------------------------------------------------------

'returns the current project, view, projection of active view, and view extent
theProject = av.GetProject
DataView = av.GetActiveDoc

GridList = List.Make                    'Create a list that will contain the list of grid themes
ThemeList = DataView.GetThemes              'Get the list of themes in the active view
If (ThemeList.Count = 0) then               'If there are not themes error message
  msgbox.info("There are no themes in the data view, please add some and run
        again","Error")   'No themes in the view to perform analysis on
else
  theProjection = DataView.GetProjection
  DataViewExtent = DataView.ReturnExtent
  for each i in 0..(ThemeList.count - 1)
    If (ThemeList.Get(i).Is(GTheme)) then      'Create a list that contains only the grid
        themes
      GridList = GridList.Add(ThemeList.Get(i))
    end
  end                                   'end for
  If (GridList.count < 2) then             'Will need at least 2 grids to get probabilities
        display an error message
   msgbox.info("There Needs to be at least 2 grids in the view","Error")
  Else
   results = av.run("Markov.SortGrid",{GridList})  'calls the script to sort the list of grids
        - it was assumed grids were named in ascending order based on year
   n = GridList.Count - 1                  'n (number of transitions) = Number of grid - 1
   av.GetProject.GetWorkDir.SetCwd
   tempgrid = Grid.MakeFromNumb(0)          'create temp. grids that will start with 0
        then contain the number of times a transition is made
   tempgrid2 = Grid.MakeFromNumb(0)
   count0 = Grid.MakeFromNumb(0)
   count1 = Grid.MakeFromNumb(0)
   m = n - 1                              'm (number of compares) = number of transitions - 1
   for each i in 0..m                     'Get the grids in pairs and compare to see if they make
        a transition from 0 to 1 and 1 to 0
     g1 = GridList.Get(i)                  'If they make a transition add 1 to the grid containing
        the sum else return the grid containing the sum
     g1Grid = g1.GetGrid
     g2 = GridList.Get(i+1)
     g2Grid = G2.GetGrid
```

```
  count0 = (g1grid = 0).con(count0 + 1.asgrid, count0)  'Count the number of
      transitions that start with 0 or 1
  count1 = (g1grid = 1).con(count1 + 1.asgrid, count1)
  tempgrid = ((g1grid = 0) and (g2grid = 1)).con(tempgrid + 1.asgrid, tempgrid)
  tempgrid2 = ((g1grid = 1) and (g2grid = 0)).con(tempgrid2 + 1.asgrid, tempgrid2)
end


tempgridtheme = gtheme.make(tempgrid)

grid01 = tempgrid.float/count0              'To get the probability divide the grid
      containing the sum by the number of grids

grid00 = 1.asgrid - grid01              'The grids add up to unity to find the opposite grid
      1 - probability found above

tempgrid2theme = gtheme.make(tempgrid2)

grid10 = tempgrid2.float/count1

grid11 = 1.asgrid - grid10


PTransView = View.Make          'Create a view that will display the transitions
      probabilities

results = av.run("Markov.SetUpGrids",{PtransView,grid00,True,"Transition From
      Uninfested to Uninfested","tranprobb1.avl","grid00theme"})  'calls the script to
      create a GTheme and add it to the view and set up its legend
results = av.run("Markov.SetUpGrids",{PtransView,grid01,True,"Transition From
      Uninfested to Infested","tranproba1.avl","grid01theme"})
results = av.run("Markov.SetUpGrids",{PtransView,grid10,True,"Transition From
      Infested to Uninfested","tranprobb.avl","grid10theme"})
results = av.run("Markov.SetUpGrids",{PtransView,grid11,True,"Transition From
      Infested to Infested","tranproba.avl","grid11theme"})

PTransViewProjection = PTransView.GetProjection         'returns the projection of the
      new view (transitions probabilities)
results = av.run("Markov.SetProj",{PTransViewProjection,PTransView})  'calls the
      script to set the project of the view
PTransViewDisplay = PTransView.GetDisplay          'Set the display of the transition
      probabilities to be the same as the data view
PTransViewDisplay.ZoomToRect(DataViewExtent)        'Zoom to same area as
      displayed in data view
```

```
   PTransView.SetComments("One-Step Markov Transition probabilities.  (Shows the 0-
         >0, 0->1, 1->0, and 1->1 transitions)")    'Set up the views comment field
   PTransView.SetName("1 Year Transition Probabilities")    'Set the Name of the view
         that contains the transitions probabilities
   PTransWindow = PTransView.GetWin
   PTransWindow.Open                              'Opens the new view window
  end         'end if (themelist.count < 2)
end            'end if (themelist.count = 0)
```

'Name: Markov.SetProj

```
'
'Title: Sets the Projection of a view
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will set the projection of a view to Lambert Conformal Conic if no projection
         is set for the view.
'
'Requires:
'  Calls:
'
'Self:(projection,view)
'Returns:
'-----------------------------------------------------------------------

theProjection = self.Get(0)
theView = self.Get(1)

If (theProjection.IsNull) then             'If the view is not projected set the projection
  aRect = Rect.MakeXY(-180,-90,180,90)      'Creates a rectangle around the entire world
  'Set the parameters of the projection
  'set the projection to Lambert Conformal Conic, spheroid to Clarke 1866, central
         meridian to -107.5538
  'set reference lat to 41, standard parallel1 to 41, standard parallel2 to 45, and easting and
         northing to 0
  p = Lambert.Make(aRect)
  p.SetSpheroid(#SPHEROID_CLARKE1866)
  p.SetCentralMeridian(-107.5538)
  p.SetReferenceLatitude(41)
  p.SetLowerStandardParallel(41)
  p.SetUpperStandardParallel(45)
  p.SetFalseEasting(0)
  p.SetFalseNorthing(0)
  theView.SetProjection(p)
end
```

```
'Name: Markov.SetProj.Tool
'
'Title: Sets the Projection of a view
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will set the projection of a view to Lambert Conformal Conic if no projection
        is set for the view.
'
'Requires:
'  Calls:
'
'Self:(projection,view)
'Returns:
'-----------------------------------------------------------------------

theProject = av.GetProject
DataView = av.GetActiveDoc
theProjection = DataView.GetProjection


If (theProjection.IsNull) then          'If the view is not projected set the projection
  aRect = Rect.MakeXY(-180,-90,180,90)    'Creates a rectangle around the entire world
  'Set the parameters of the projection
  'set the projection to Lambert Conformal Conic, spheroid to Clarke 1866, central
        meridian to -107.5538
  'set reference lat to 41, standard parallel1 to 41, standard parallel2 to 45, and easting and
        northing to 0
  p = Lambert.Make(aRect)
  p.SetSpheroid(#SPHEROID_CLARKE1866)
  p.SetCentralMeridian(-107.5538)
  p.SetReferenceLatitude(41)
  p.SetLowerStandardParallel(41)
  p.SetUpperStandardParallel(45)
  p.SetFalseEasting(0)
  p.SetFalseNorthing(0)
  DataView.SetProjection(p)
end


'Name: Markov.SetUpGrids
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
```

'This Script will create a GTheme, add the theme to a view, Set the theme to visible,
'set the name of the theme, get the legend and apply a legend.
'
'Requires:
'
'   Calls: Markov.CreateLegends
'
'Self: (View, Grid, Visible Boolean, Grid Theme Name, Legend file)
'Returns:
'---

```
TheView = self.Get(0)
TheGrid = self.Get(1)
BolVisible = self.Get(2)
GridName = self.Get(3)
LegendFile = self.Get(4)
theGridTheme = self.get(5)

TheGridTheme = GTheme.Make(TheGrid)
TheView.AddTheme(TheGridTheme)
If (BolVisible) then
  TheGridTheme.SetVisible(True)
End
TheGridTheme.SetName(GridName)
TheGridThemeLegend = TheGridTheme.GetLegend
results =
        av.run("Markov.CreateLegends",{TheGridThemeLegend,LegendFile,#LEGEND
        _LOADTYPE_ALL})  'calls the script to apply a legend to the new theme
```

'Name: SortData

'
'Author: Kiana Zimmerman
'---
'This Script will sort the grids in the data view in ascending order
'
'Requires:
'   Calls:
'
'Self: (Grid list)
'Returns:
'---

```
GridList = self.Get(0)

n = GridList.Count                  'number of Grids in the list of Grids
```

```
for each i in 0..(n-2)            'Goes through the list of themes and finds the sorted order
  BestYet = GridList.Get(i)
  BestYetIndex = i
  For each j in (i+1)..(n-1)
    if (GridList.Get(j).GetName < BestYet.GetName) then
      BestYet = GridList.Get(j)
      BestYetIndex = j
    end                         'if (GridList.Get(j).GetName < BetYet.GetName)
  end                           'end j loop
  GridList.Set(BestYetIndex,GridList.Get(i))  'Puts the best choice Grid in the best index
  GridList.Set(i,BestYet)
end                             'end i loop
```

'Name: Markov.U00

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will Compute next n years infestation Uninfested starting with Uninfested
'
'Requires:
'    Calls:
'
'Self: (a grid, b grid, k)
'Returns: U00 grid
'-----------------------------------------------------------------------

aGrid = self.Get(0)
bGrid = self.Get(1)
k = self.Get(2)

return ((bGrid/(agrid + bgrid)) + ((agrid * ((1.AsGrid - agrid - bgrid).Pow(k)))/(agrid +
        bgrid)))
```

'Name: Markov.U01

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will compute next n years infestation infested starting with Uninfested
'
'Requires:
'    Calls:
'
'Self: (a grid, b grid, k)
'Returns: U01 grid
```

```
'-----------------------------------------------------------------------

aGrid = self.Get(0)
bGrid = self.Get(1)
k = self.Get(2)

return ((aGrid/(agrid + bgrid)) - ((agrid * ((1.AsGrid - agrid - bgrid).Pow(k)))/(agrid +
        bgrid)))
```

'Name: Markov.U10

```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will compute next n years infestation Uninfested starting with infested
'
'Requires:
'   Calls:
'
'Self: (a grid, b grid, k)
'Returns: U10 grid
'-----------------------------------------------------------------------

aGrid = self.Get(0)
bGrid = self.Get(1)
k = self.Get(2)

return ((bGrid/(agrid + bgrid)) - ((bgrid * ((1.AsGrid - agrid - bgrid).Pow(k)))/(agrid +
        bgrid)))
```

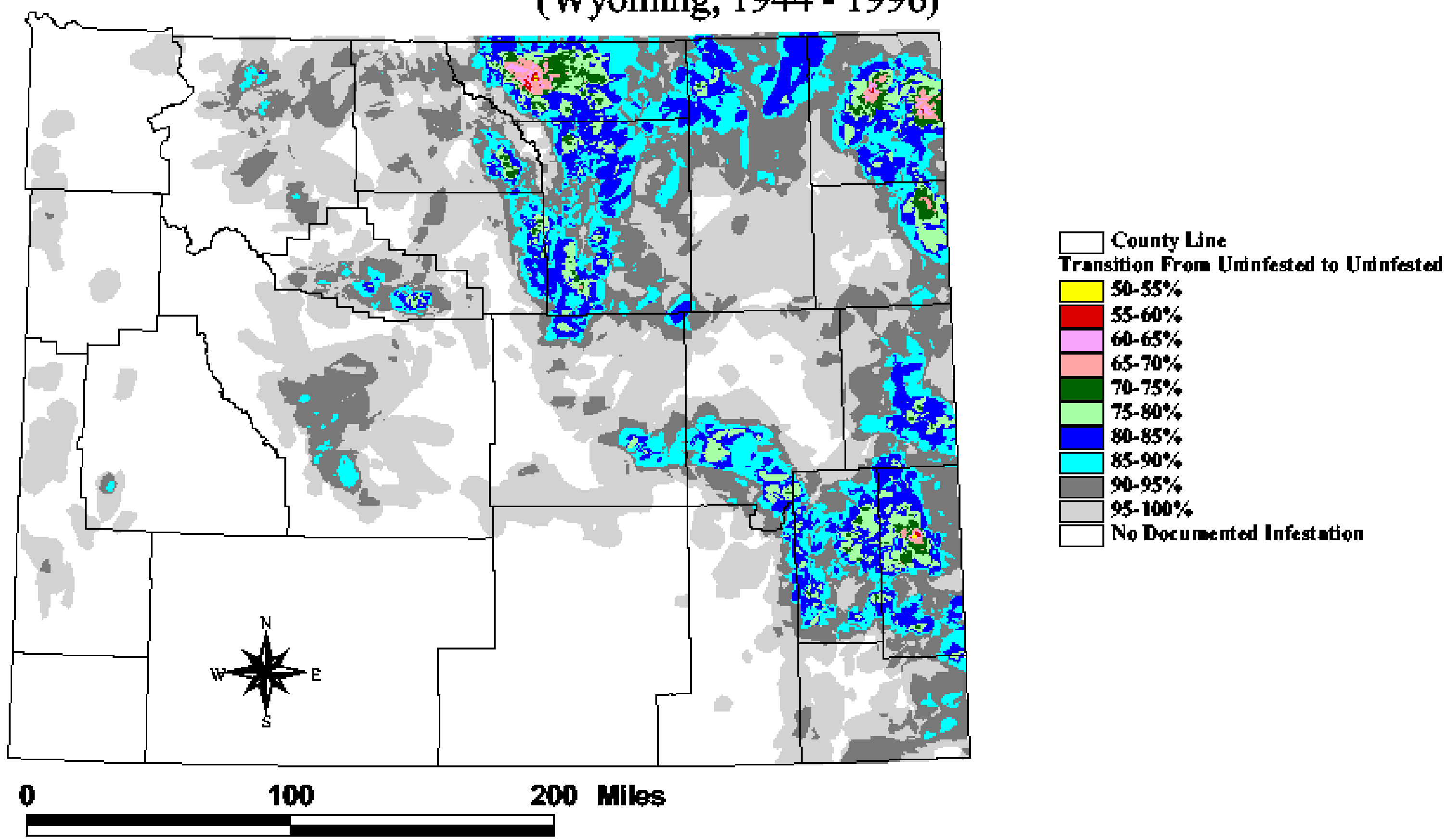
'Name: Markov.U11

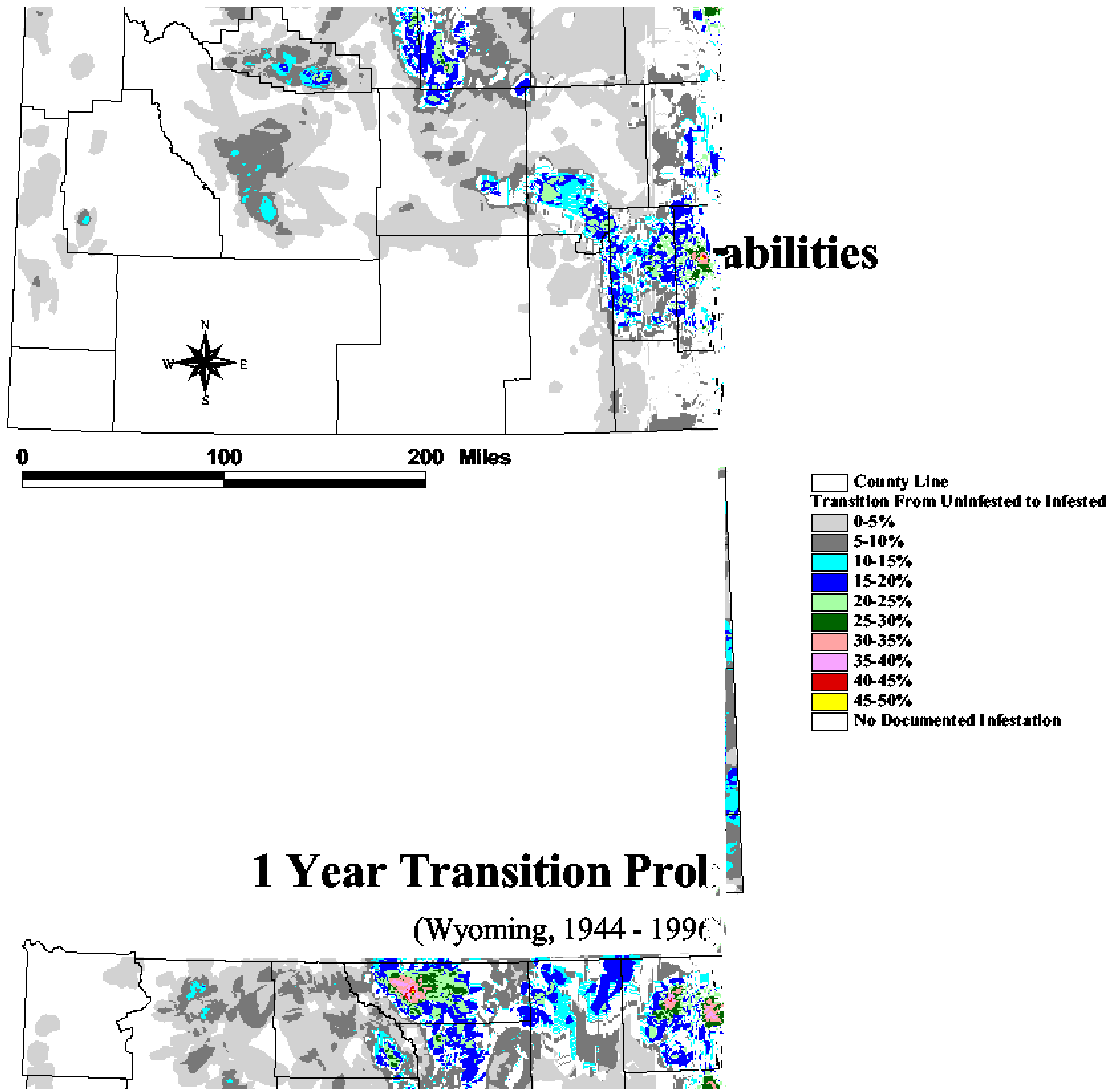
```
'
'Author: Kiana Zimmerman
'-----------------------------------------------------------------------
'This Script will compute next n years infestation Infested starting with Infested
'
'Requires:
'   Calls:
'
'Self: (a grid, b grid, k)
'Returns: U11 grid
'-----------------------------------------------------------------------

aGrid = self.Get(0)
bGrid = self.Get(1)
```

Appendix B

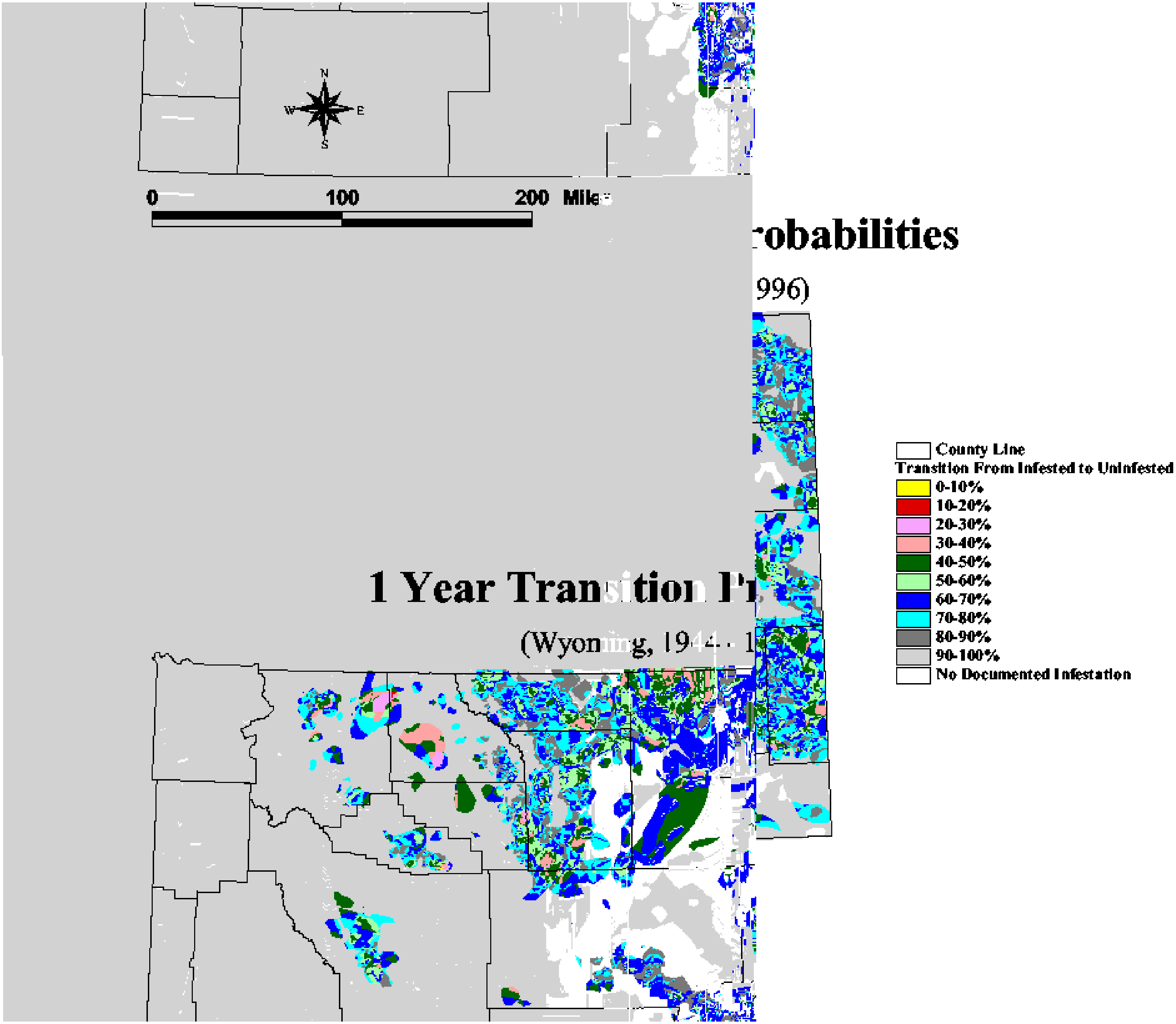
1 Year Transition Probabilities

(Wyoming, 1944 - 1996)



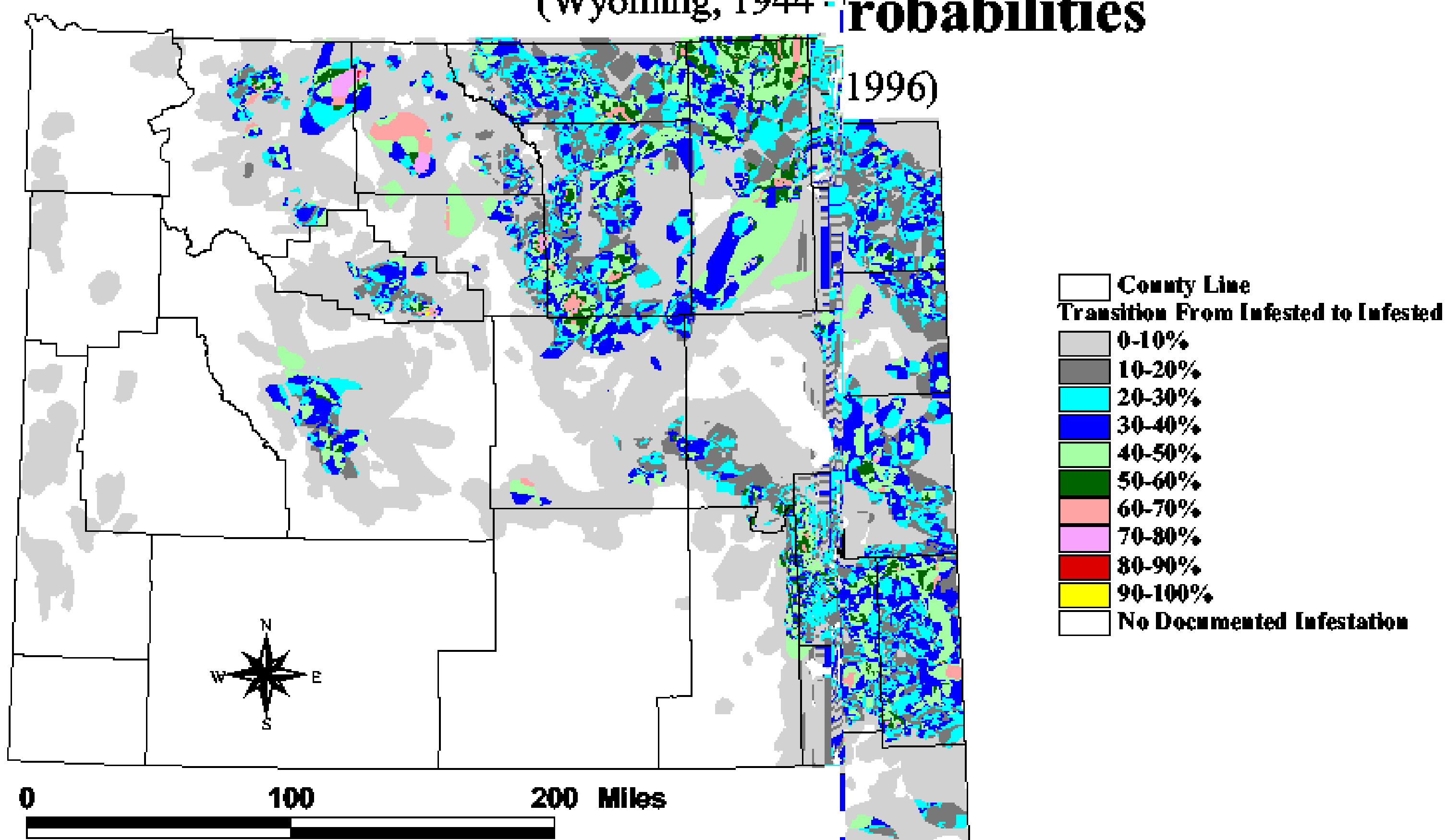


abilities
N
W
E
S
0
100
200 Miles
County Line
Transition From Uninfested to Infested
0-5%
5-10%
10-15%
15-20%
20-25%
25-30%
30-35%
35-40%
40-45%
45-50%
No Documented Infestation
1 Year Transition Prol
(Wyoming, 1944 - 199


1 Year Tran tion Pr
robabilities
(Wyon g, 19 4 1
996)
0
100
200 Miles
N
S
E
W
County Line
Transition From Infested to Uninfested
0-10%
10-20%
20-30%
30-40%
40-50%
50-60%
60-70%
70-80%
80-90%
90-100%
No Documented Infestation

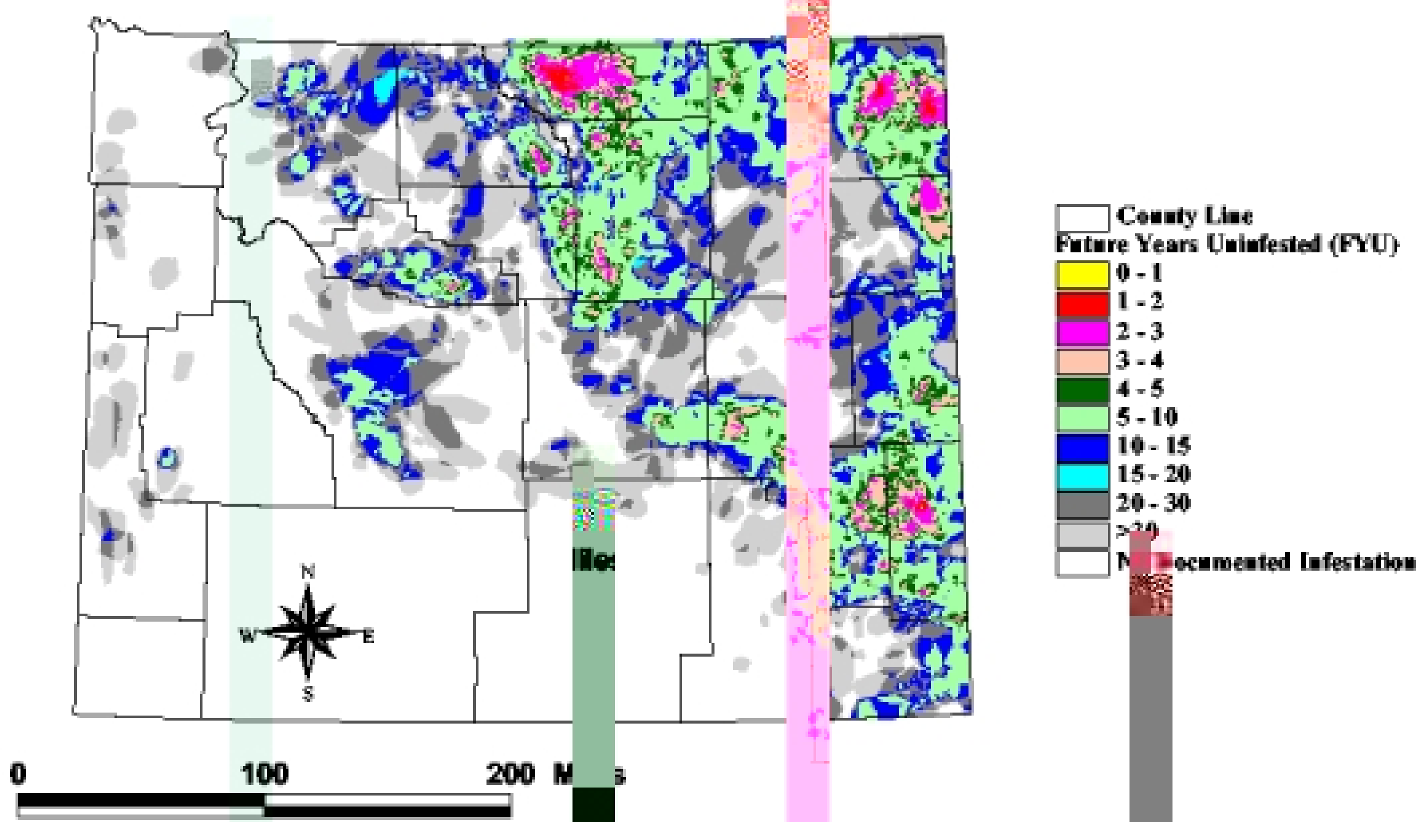
1 Year Transition Probabilities

(Wyoming, 1944 - 1996)



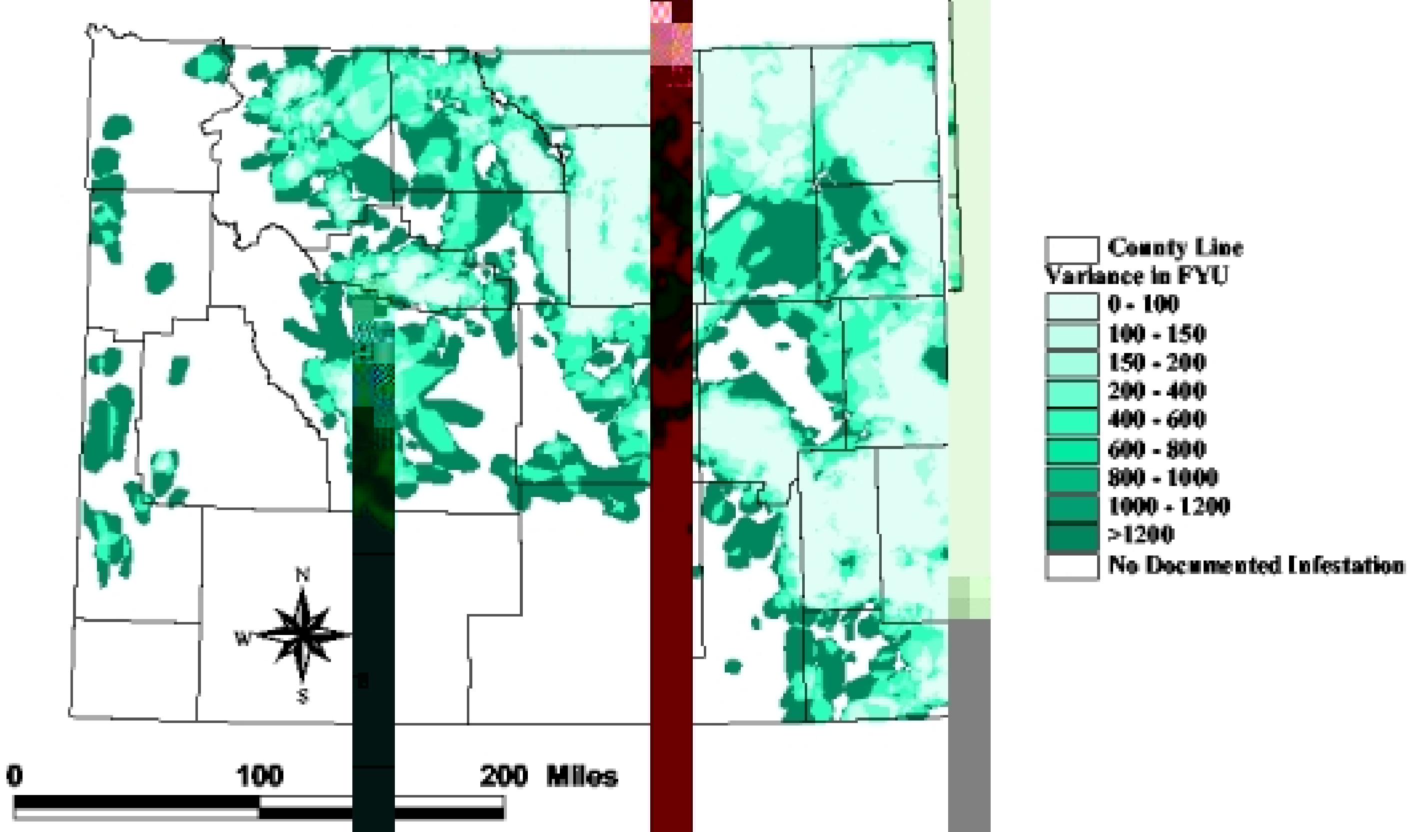
Expected Number of Years

(Wyoming, 1944 - 1996)



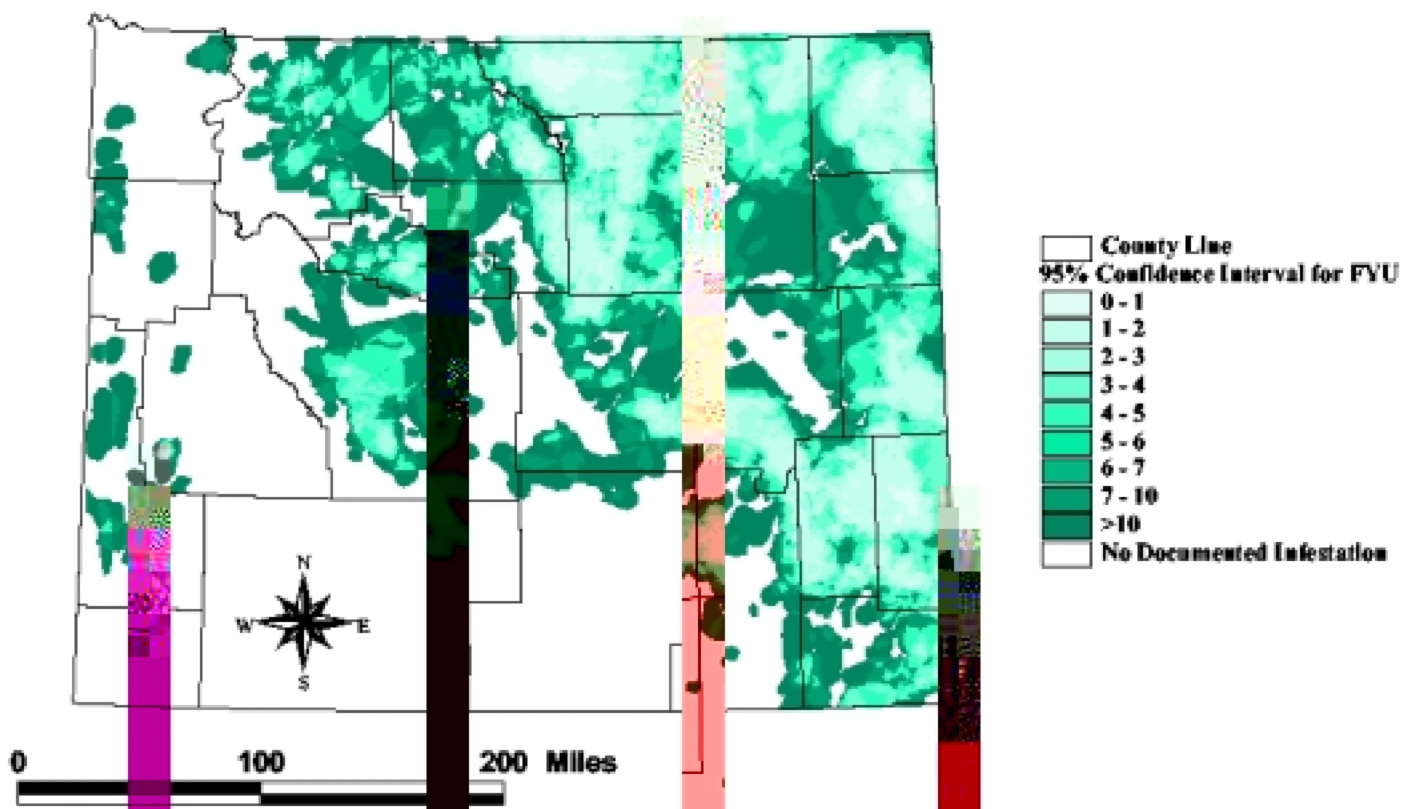
Expected Number of Years

(Wyoming, 1944 - 1996)



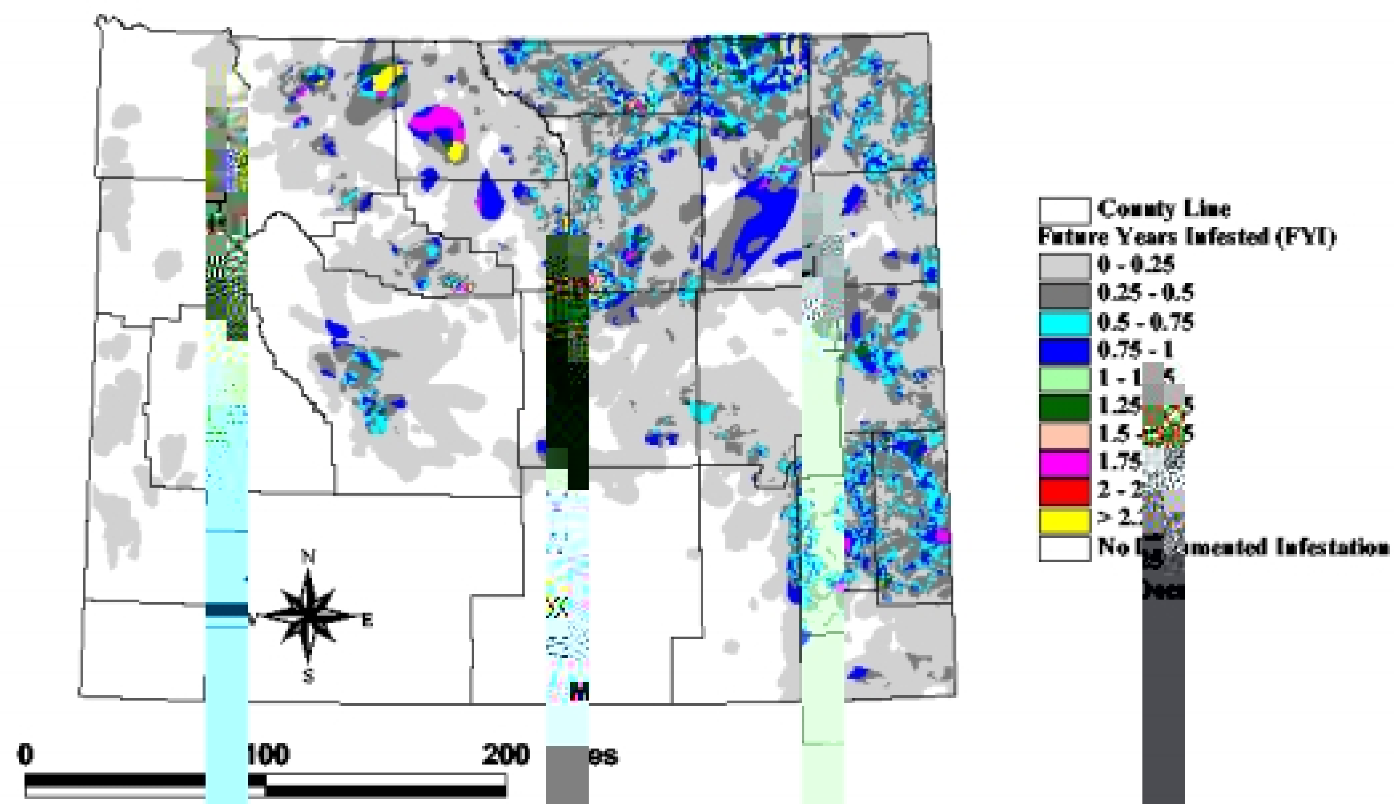
Expected Number of Years

(Wyoming, 1944 - 1996)



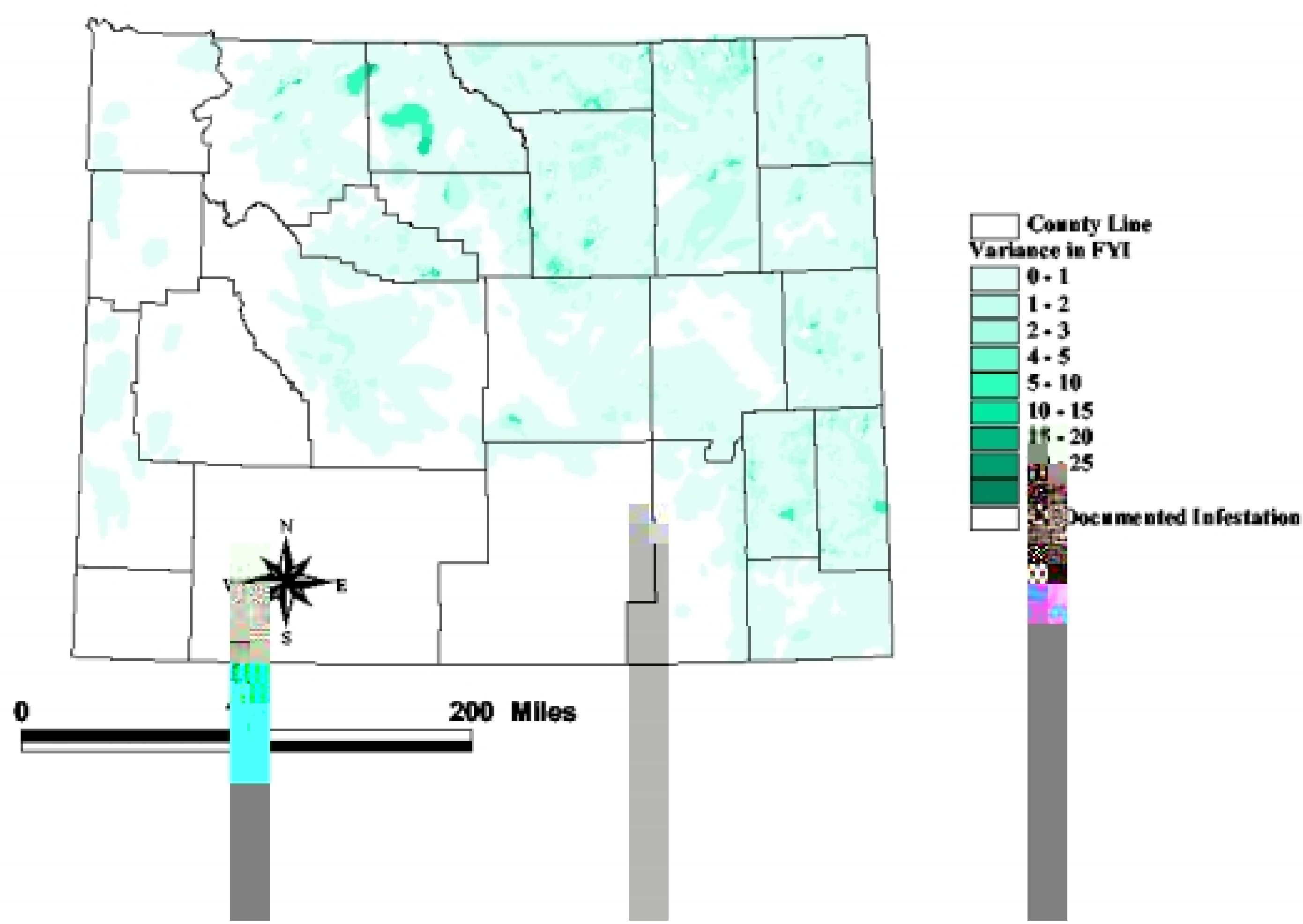
Expected Number of Years

(Wyoming, 1944 - 1996)



Expected Number of Years

(Wyoming, 1944 - 1996)

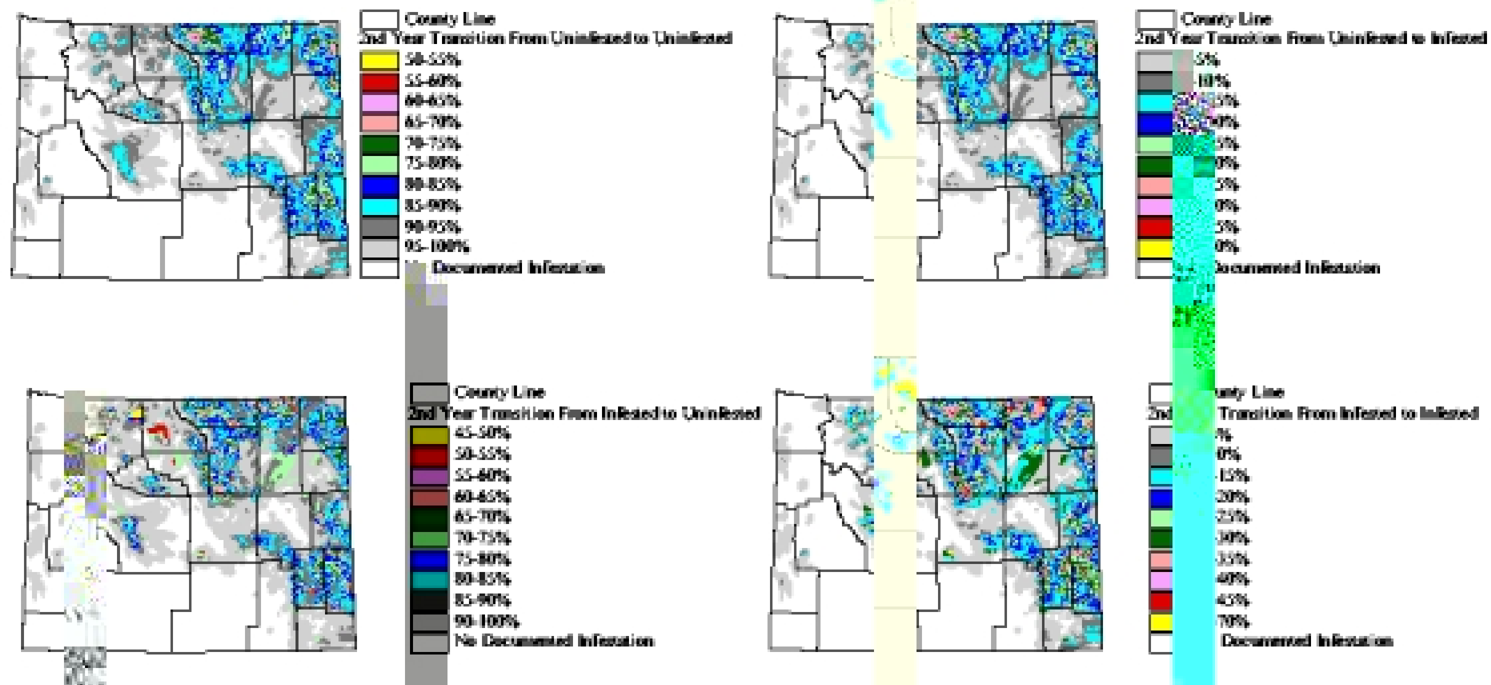


for FYI

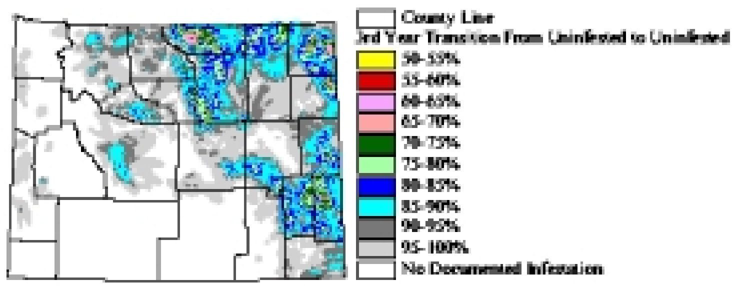
station

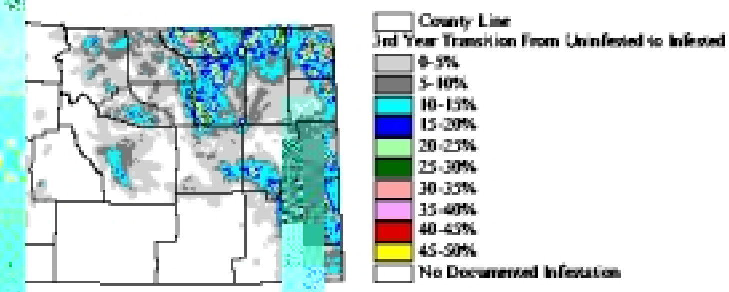
[redacted]

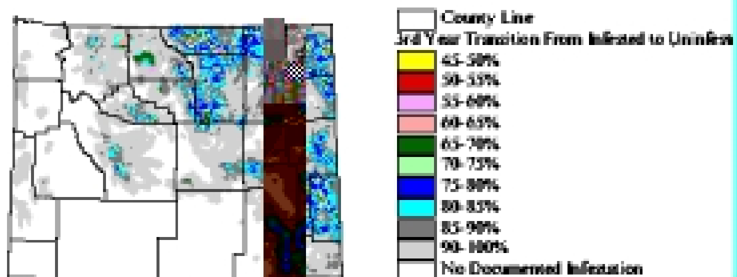
nth-Step Transitions
n = 2
(Wyoming, 1944 - 1996)

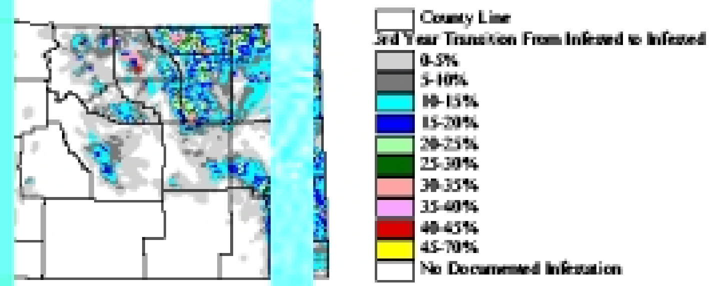


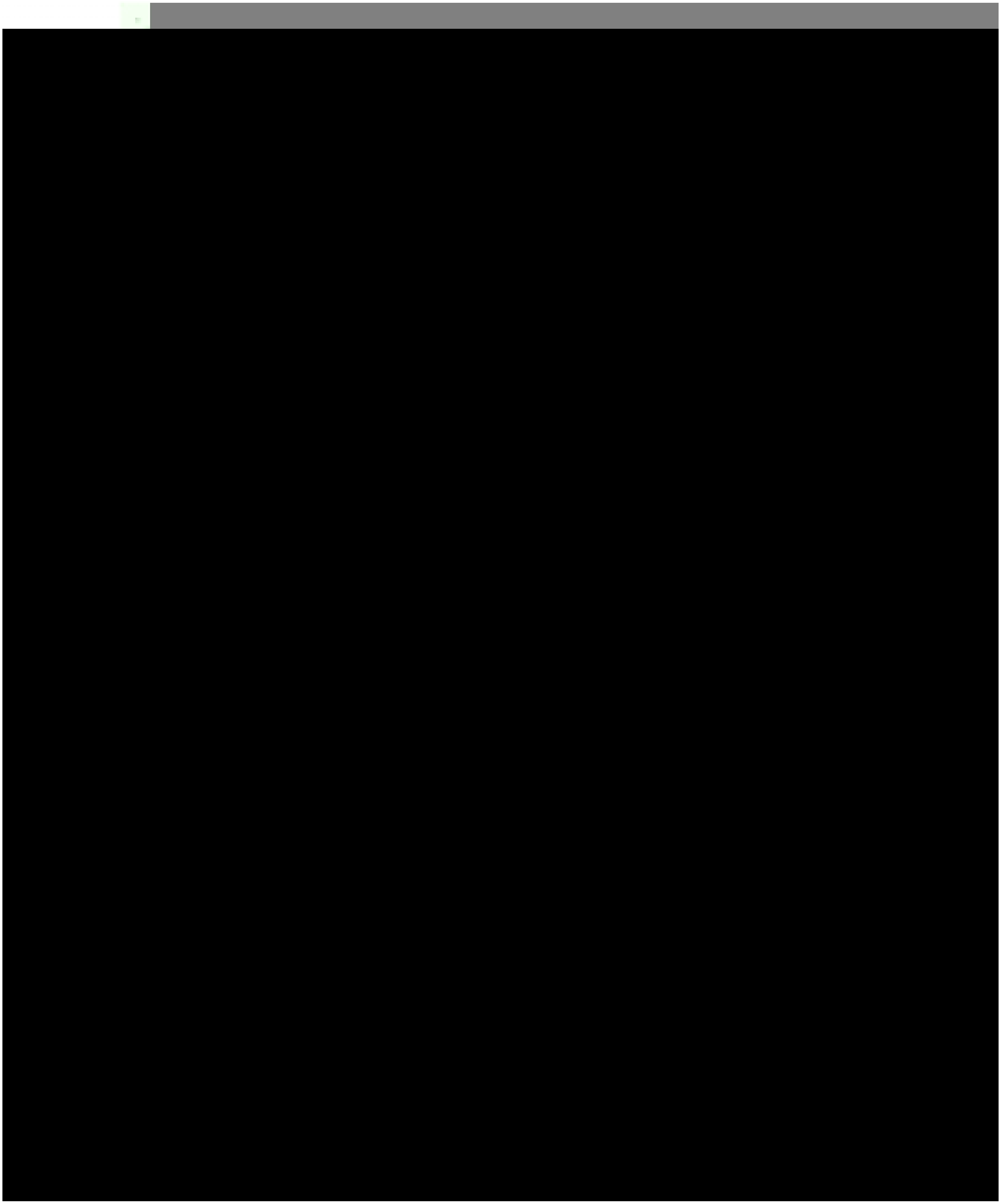
nth-Step Transitions
n = 3
(Wyoming, [illegible]44 - 1996)



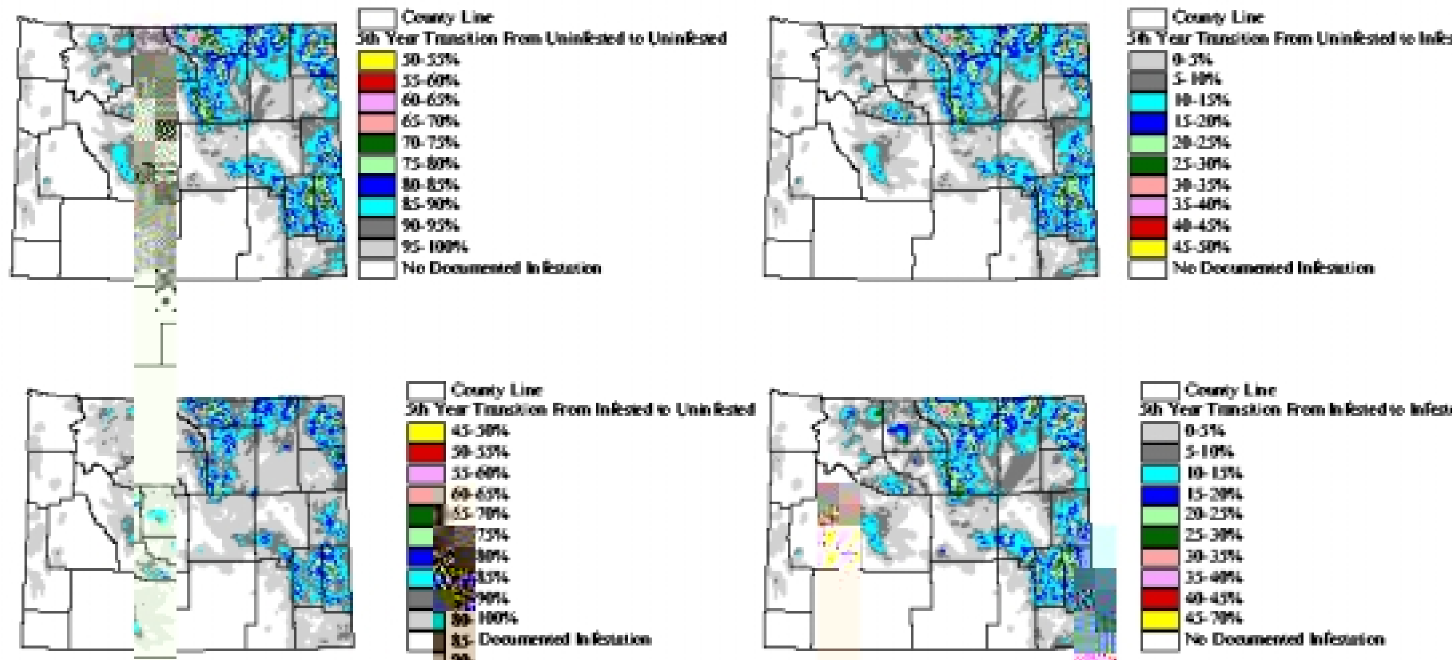






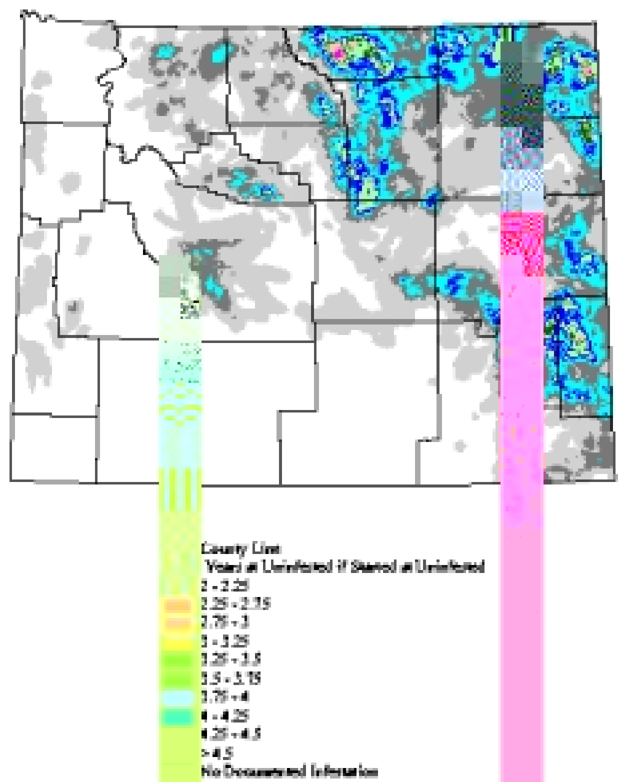


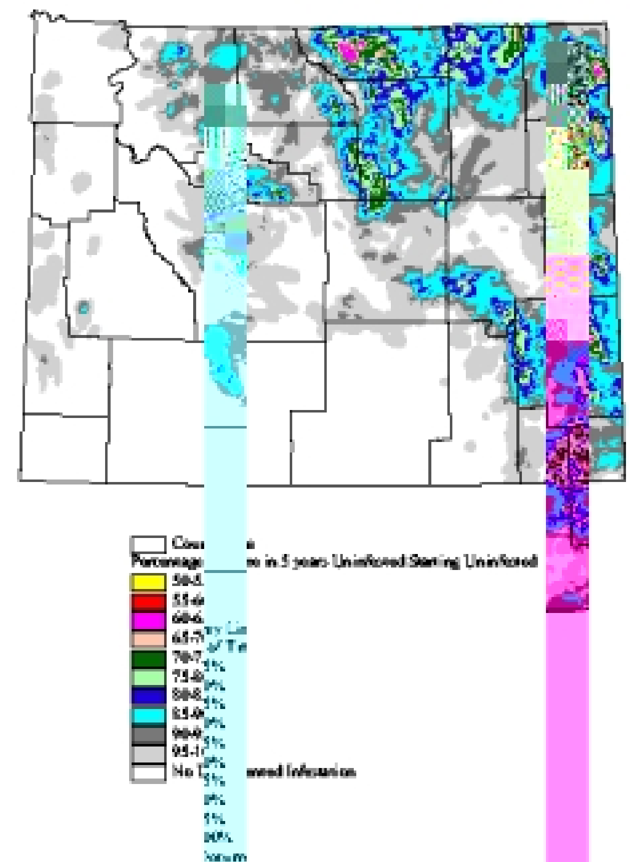
nth-Step Transitions
n = 5
(Wyoming, 1944 - 1996)



Expected Number of Future Years in Each State

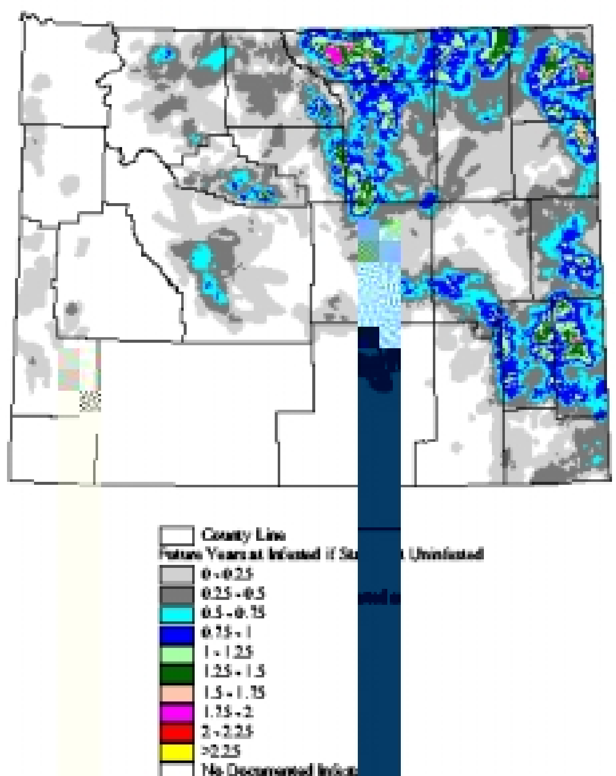
(Wyoming, 1944 - 1996)

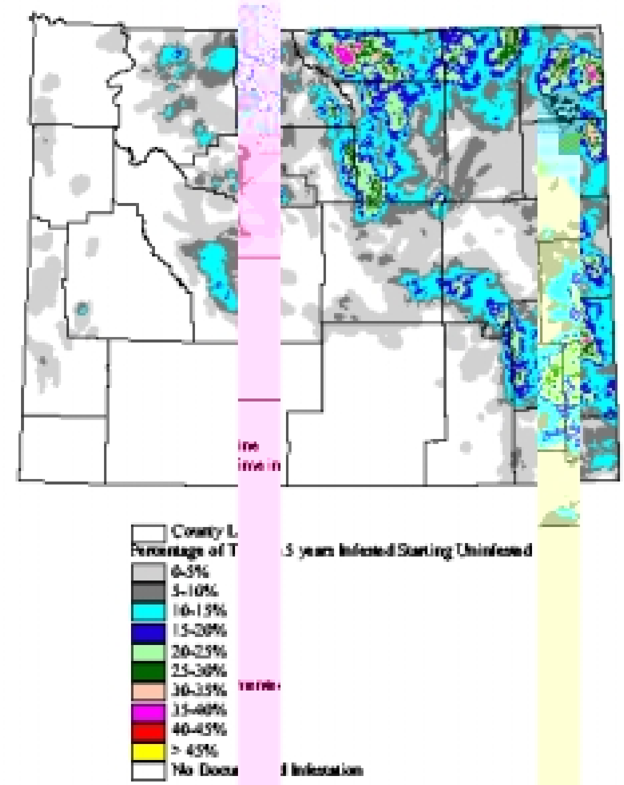




Expected Number of Future Years in Each State

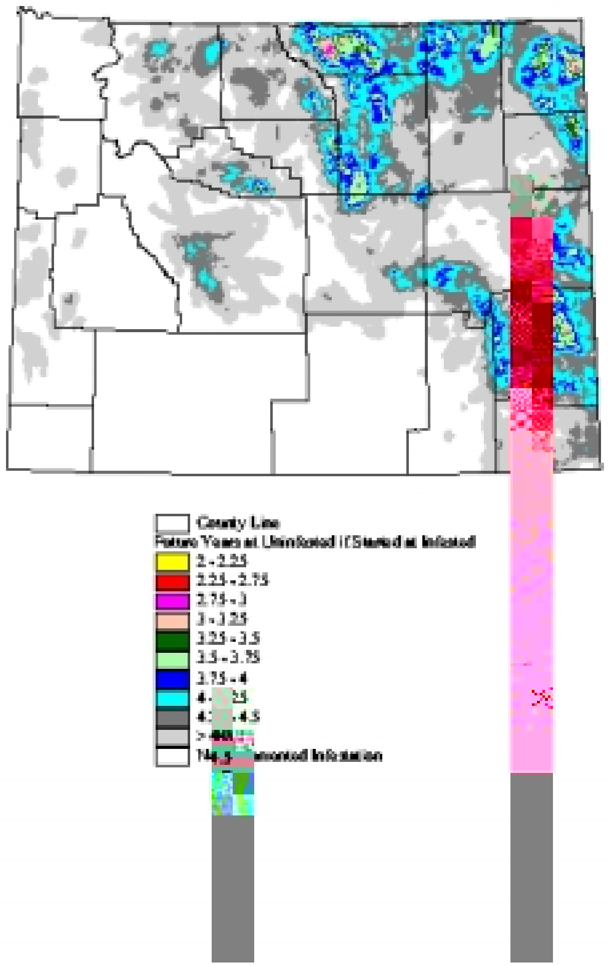
(Wyoming, 1944 - 1996)

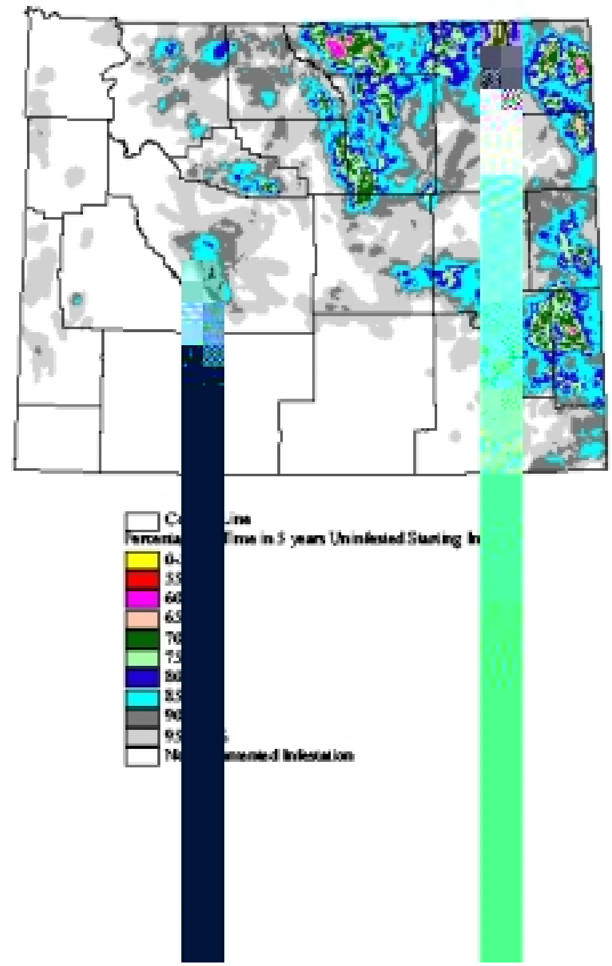




Expected Number of Future Years in Each State

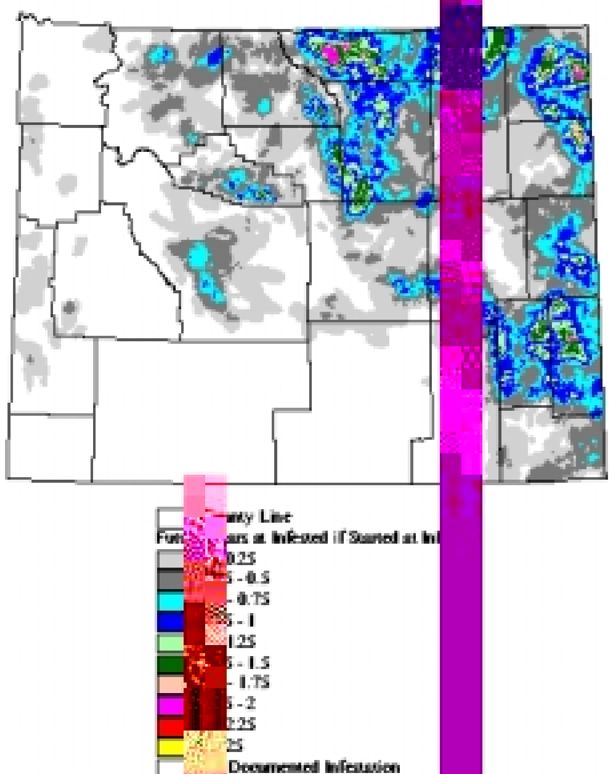
(Wyoming, 1944 - 1996)

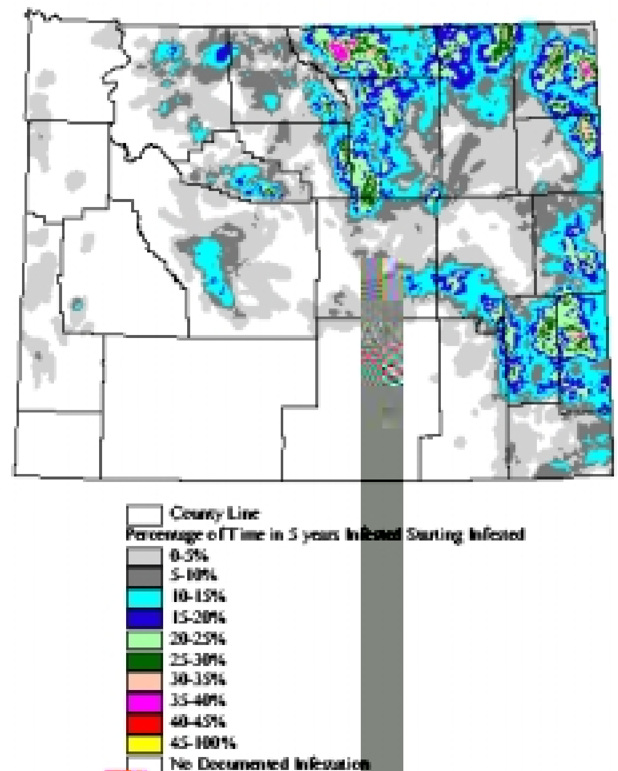




Expected Number of Future Years in Each State

(Wyoming, 1944 - 1996)





Outbreak Frequency

(Wyoming, 1944 - 1996)